

(19) 世界知的所有権機関
国際事務局



(43) 国際公開日
2001 年 11 月 1 日 (01.11.2001)

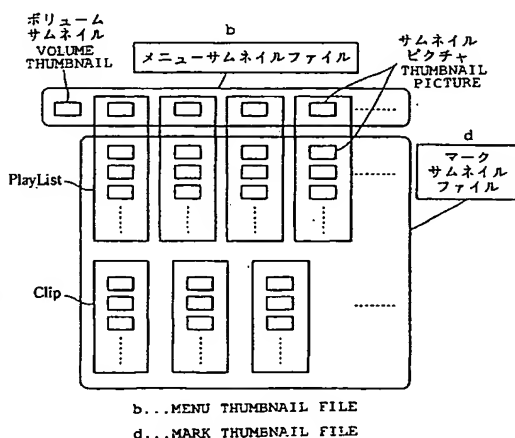
PCT

(10) 国際公開番号
WO 01/82608 A1

- (51) 国際特許分類: H04N 5/93, G11B 20/10 Motoki) [JP/JP]. 浜田俊也 (HAMADA, Toshiya) [JP/JP]; 〒141-0001 東京都品川区北品川6丁目7番35号 ソニー株式会社内 Tokyo (JP).
- (21) 国際出願番号: PCT/JP01/03414
- (22) 国際出願日: 2001 年 4 月 20 日 (20.04.2001) (74) 代理人: 小池 晃, 外 (KOIKE, Akira et al.); 〒105-0001 東京都港区虎ノ門二丁目6番4号 第11森ビル Tokyo (JP).
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語 (81) 指定国 (国内): CN, KR, US.
- (30) 優先権データ: (84) 指定国 (広域): ヨーロッパ特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).
特願2000-183770 2000 年 4 月 21 日 (21.04.2000) JP
特願2000-268043 2000 年 9 月 5 日 (05.09.2000) JP
- (71) 出願人 (米国を除く全ての指定国について): ソニー株式会社 (SONY CORPORATION) [JP/JP]; 〒141-0001 東京都品川区北品川6丁目7番35号 Tokyo (JP).
添付公開書類:
— 国際調査報告書
— 補正書
- (72) 発明者; および
(75) 発明者/出願人 (米国についてののみ): 加藤元樹 (KATO, 2 文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

(54) Title: INFORMATION PROCESSING APPARATUS AND METHOD, PROGRAM, AND RECORDED MEDIUM

(54) 発明の名称: 情報処理装置及び方法、プログラム、並びに記録媒体



(57) Abstract: A ClipMark composed of a mark indicating a characteristic image extracted from an inputted AV stream is created as management information for managing the AV stream. A PlayListMark composed of a mark indicating an image arbitrarily specified by the user is created from a reproduction section corresponding to a PlayList defining a combination of predetermined sections in the AV stream. The ClipMark and PlayListMark are recorded as mutually independent tables on a recording medium. Therefore a quick and reliable access to a desired position of the AV stream can be made.

WO 01/82608 A1



(57) 要約:

入力されたA Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、A Vストリームを管理するための管理情報として生成すると共に、A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成し、ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録するようにしたので、A Vストリームの所望の位置に、迅速且つ確実にアクセスすることが可能となる。

明細書

情報処理装置及び方法、プログラム、並びに記録媒体

技術分野

本発明は情報処理装置及び方法、プログラム、並びに記録媒体に関し、特に、A Vストリームの所望の位置に、迅速にアクセスすることができるようにした情報処理装置及び方法、プログラム、並びに記録媒体に関する。

背景技術

近年、記録可能で記録再生装置から取り外し可能なディスク型媒体として、各種の光ディスクが提案されている。このような記録可能な光ディスクは、数ギガバイトの大容量メディアとして提案されており、ビデオ信号等のA V (Audio Visual) 信号を記録するメディアとしての期待が高い。

この記録可能な光ディスクに記録するデジタルのA V信号のソース（供給源）としては、記録装置自身が、アナログ入力のオーディオビデオ信号を、MPEG-2方式で画像圧縮して作るビットストリームや、デジタルテレビジョン放送の電波から直接得られるMPEG-2方式のビットストリームなどがある。一般に、デジタルテレビジョン放送では、MPEG-2トランスポートストリームが使われる。トランスポートストリームは、トランスポートパケットが連続したストリームであり、トランスポートパケットは、例えば、MPEG-2ビデオストリームやMPEG-1オーディオストリームがパケット化されたものである。1つのトランスポートパケットのデータ長は188バイトである。デジタルテレビジョン放送で受信されるトランスポートストリームのA Vプログラムを記録装置で光ディスクにそのまま記録すれば、ビデオやオーディオの品質を全く劣化させることなく記録することが可能である。

ユーザが、光ディスクに記録されているトランスポートストリームの中から興

味のあるシーン、例えば番組の頭出し点などをサーチできるようにするために、再生装置はランダムアクセス再生ができることが求められる。

一般に、MPEG-2ビデオのストリームは、0.5秒程度の間隔でIピクチャを符号化し、それ以外のピクチャはPピクチャ又はBピクチャとして符号化される。したがって、MPEG-2ビデオのストリームが記録された光ディスクから、ランダムアクセスし、ビデオ再生する場合、はじめに、Iピクチャをサーチしなければならない。

しかしながら、従来は、光ディスクに記録されているトランスポートストリームに、ランダムアクセスし、ビデオ再生する場合に、Iピクチャの開始バイトを効率良くサーチすることが困難であった。すなわち、光ディスク上のトランスポートストリームのランダムなバイト位置から、読み出したビデオストリームのシンタクスを解析し、Iピクチャの開始バイトをサーチしなければならず、Iピクチャのサーチに時間がかかり、ユーザからの入力に対して応答の速いランダムアクセス再生を行うことが困難であった。

発明の開示

本発明の目的は、このような状況を鑑み、ユーザのランダムアクセス再生の指示に対して、記録媒体からのトランスポートストリームの読出位置の決定とストリームの復号開始を速やかに行えるようにすることにある。

本発明に係る情報処理装置は、入力されたAVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、AVストリームを管理するための管理情報として生成すると共に、AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成手段と、ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する記録手段とを有する。

ここで、生成手段は、ClipMarkをClipMarkInformationファイルとして生成すると共に、PlayListをPlayListファイルとして生成することができる。

PlayListMarkは、PlayListを再生するときのResume点を示すマークを更に含むようにすることができる。

PlayListを再生するとき、PlayListの再生区間に対応するA VストリームのClipMarkを構成するマークを参照するようにすることができる。

PlayListMarkのマークは、プレゼンテーションタイムスタンプと、PlayListの再生経路を構成するA Vストリームデータ上の指定された1つの再生区間を示す識別情報を含むようにすることができる。

ClipMarkを構成するマーク、又は、PlayListMarkを構成するマークは、エントリーストリームのエントリポイントを特定する情報を含むようにすることができる。

PlayListMarkのマークは、ユーザが指定したお気に入りのシーンの開始点又はPlayListのResume点を少なくとも含むタイプの情報を含むようにすることができる。

ClipMarkを構成するマークとPlayListMarkを構成するマークは、A Vストリームのエントリポイントに対応する相対的なソースパケットのアドレスで表されるようにすることができる。

ClipMarkを構成するマークとPlayListMarkを構成するマークは、A Vストリームのエントリポイントに対応する相対的なソースパケットの第1のアドレスと、第1のアドレスからのオフセットのアドレスである第2のアドレスで表されるようにすることができる。

第1の記録手段による記録の際に検出された特徴的な画像のタイプを検出するタイプ検出手段を更に含み、第1の記録手段は、ClipMarkを構成するマークと、タイプ検出手段により検出されたタイプとを対応させて記録するようにすることができる。

ClipMarkのマークは、シーンチェンジ点、コマーシャルの開始点、コマーシャルの終了点、又はタイトルが表示されたシーンを含むようにすることができる。

本発明に係る情報処理方法は、入力されたA Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、A Vストリームを管理するための管理情報として生成すると共に、A Vストリーム中の所定の区間の組み合

わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成ステップと、ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御ステップとを有する。

本発明に係る記録媒体のプログラムは、入力されたAVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、AVストリームを管理するための管理情報として生成すると共に、AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成ステップと、ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御ステップとを含む。

本発明に係るプログラムは、入力されたAVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、AVストリームを管理するための管理情報として生成すると共に、AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成ステップと、ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御ステップとをコンピュータに実行させる。

本発明の情報処理装置は、AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含むAVストリームを管理するための管理情報と、AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを読み出す読出手段と、読出手段により読み出された管理情報とPlayListMarkによる情報を提示する提示手段と、提示手段により提示された情報から、ユーザが再生を指示したPlayListに対応するClipMarkを参照する参照手段と、参照手段により参照されたClipMarkを含み、ClipMarkに対応する位置からAVストリームを再生する再生手段とを含む。

ここで、提示手段は、PlayListMarkに対応するサムネイル画像によるリストをユーザに提示するようにすることができる。

ClipMarkを構成するマークとPlayListMarkを構成するマークは、A Vストリームのエントリポイントに対応する相対的なソースパケットのアドレスで表されるようにすることができる。

ClipMarkを構成するマークとPlayListMarkを構成するマークは、A Vストリームのエントリポイントに対応する相対的なソースパケットの第1のアドレスと、第1のアドレスからのオフセットのアドレスである第2のアドレスで表されるようにすることができる。

ClipMarkのマークは、シーンチェンジ点、コマーシャルの開始点、コマーシャルの終了点、又はタイトルが表示されたシーンを含むようにすることができる。

本発明に係る情報処理装置は、A Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含むA Vストリームを管理するための管理情報と、A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkの読み出しを制御する読出制御ステップと、読出制御ステップの処理で読み出しが制御された管理情報とPlayListMarkによる情報を提示する提示ステップと、提示ステップの処理で提示された情報から、ユーザが再生を指示したPlayListに対応するClipMarkを参照する参照ステップと、参照ステップの処理で参照されたClipMarkを含み、ClipMarkに対応する位置からのA Vストリームの再生を制御する再生制御ステップとを含む。

本発明に係る記録媒体のプログラムは、A Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含むA Vストリームを管理するための管理情報と、A Vストリーム中の所定の区間の組み合わせを定義するPlay Listに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkの読み出しを制御する読出制御ステップと、読出制御ステップの処理で読み出しが制御された管理情報とPlayListMarkによる情報を提示する提示ステップと、提示ステップの処理で提示された情報から、ユーザが再生を指示したPlayListに対応するClipMarkを参照する参照ステップと、参照ステップの処理で参照されたClipMarkを含み、ClipMarkに対応する位置からのA Vストリームの再生を制御する再生制御ステップとを含む。

本発明に係るプログラムは、A Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含むA Vストリームを管理するための管理情報と、A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkの読み出しを制御する読出制御ステップと、読出制御ステップの処理で読み出しが制御された管理情報とPlayListMarkによる情報を提示する提示ステップと、提示ステップの処理で提示された情報から、ユーザが再生を指示したPlayListに対応するClipMarkを参照する参照ステップと、参照ステップの処理で参照されたClipMarkを含み、ClipMarkに対応する位置からのA Vストリームの再生を制御する再生制御ステップとをコンピュータに実行させる。

本発明に係る記録媒体には、A Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含むA Vストリームを管理するための管理情報と、A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが、各々独立したテーブルとして記録されている。

本発明に係る情報処理装置及び方法、並びにプログラムにおいては、入力されたA Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、A Vストリームを管理するための管理情報として生成すると共に、A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが生成され、ClipMark、及びPlayListMarkが各々独立したテーブルとして記録媒体に記録される。

本発明に係る情報処理装置及び方法、並びにプログラムは、A Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含むA Vストリームを管理するための管理情報と、A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが読み出され、その読み出された管理情報とPlayListMarkによる情報が提示され、提示された情報から、ユーザが再生を指示したPlayListに対応するClipMarkが参照され、参照されたClipMark

を含み、ClipMarkに対応する位置からA Vストリームが再生される。

本発明の更に他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

図面の簡単な説明

図1は、本発明を適用した記録再生装置の一例を説明するブロック図である。

図2は、記録再生装置により記録媒体に記録されるデータのフォーマットについて説明する図である。

図3は、Real PlayListとVirtual PlayListについて説明する図である。

図4A～図4Cは、Real PlayListの作成について説明する図である。

図5A～図5Cは、Real PlayListの削除について説明する図である。

図6A及び図6Bは、アセンブル編集について説明する図である。

図7は、Virtual PlayListにサブバスを設ける場合について説明する図である。

図8は、PlayListの再生順序の変更について説明する図である。

図9は、PlayList上のマークとClip上のマークについて説明する図である。

図10は、メニューサムネイルについて説明する図である。

図11は、PlayListに付加されるマークについて説明する図である。

図12は、クリップに付加されるマークについて説明する図である。

図13は、PlayList、Clip、サムネイルファイルの関係について説明する図である。

図14は、ディレクトリ構造について説明する図である。

図15は、info.dvrのシンタクスを示す図である。

図16は、DVR volumeのシンタクスを示す図である。

図17は、Resumevolumeのシンタクスを示す図である。

図18は、UIAppInfovolumeのシンタクスを示す図である。

図19は、Character set valueのテーブルを示す図である。

図20は、TableOfPlayListのシンタクスを示す図である。

図21は、TableOfPlayListの他のシンタクスを示す図である。

図 2 2 は、MakersPrivateDataのシンタクスを示す図である。

図 2 3 は、xxxxx.rplsとyyyyy.vplsのシンタクスを示す図である。

図 2 4 A～図 2 4 C は、PlayListについて説明する図である。

図 2 5 は、PlayListのシンタクスを示す図である。

図 2 6 は、PlayList_typeのテーブルを示す図である。

図 2 7 は、UIAppinfoPlayListのシンタクスを示す図である。

図 2 8 A～図 2 8 C は、図 2 7 に示したUIAppinfoPlayListのシンタクス内のフラグについて説明する図である。

図 2 9 は、PlayItemについて説明する図である。

図 3 0 は、PlayItemについて説明する図である。

図 3 1 は、PlayItemについて説明する図である。

図 3 2 は、PlayItemのシンタクスを示す図である。

図 3 3 は、IN_timeについて説明する図である。

図 3 4 は、OUT_timeについて説明する図である。

図 3 5 は、Connection_Conditionのテーブルを示す図である。

図 3 6 A～図 3 6 D は、Connection_Conditionについて説明する図である。

図 3 7 は、BridgeSequenceInfoを説明する図である。

図 3 8 は、BridgeSequenceInfoのシンタクスを示す図である。

図 3 9 は、SubPlayItemについて説明する図である。

図 4 0 は、SubPlayItemのシンタクスを示す図である。

図 4 1 は、SubPath_typeのテーブルを示す図である。

図 4 2 は、PlayListMarkのシンタクスを示す図である。

図 4 3 は、Mark_typeのテーブルを示す図である。

図 4 4 は、Mark_time_stampを説明する図である。

図 4 5 は、zzzzz.Clipのシンタクスを示す図である。

図 4 6 は、ClipInfoのシンタクスを示す図である。

図 4 7 は、Clip_stream_typeのテーブルを示す図である。

図 4 8 は、offset_SPNについて説明する図である。

図 4 9 は、offset_SPNについて説明する図である。

図 5 0 A 及び図 5 0 B は、S T C 区間について説明する図である。

図 5 1 は、STC_Info について説明する図である。

図 5 2 は、STC_Info のシンタクスを示す図である。

図 5 3 は、ProgramInfo を説明する図である。

図 5 4 は、ProgramInfo のシンタクスを示す図である。

図 5 5 は、VideoCondngInfo のシンタクスを示す図である。

図 5 6 は、Video_format のテーブルを示す図である。

図 5 7 は、frame_rate のテーブルを示す図である。

図 5 8 は、display_aspect_ratio のテーブルを示す図である。

図 5 9 は、AudioCondngInfo のシンタクスを示す図である。

図 6 0 は、audio_coding のテーブルを示す図である。

図 6 1 は、audio_component_type のテーブルを示す図である。

図 6 2 は、sampling_frequency のテーブルを示す図である。

図 6 3 は、CPI について説明する図である。

図 6 4 は、CPI について説明する図である。

図 6 5 は、CPI のシンタクスを示す図である。

図 6 6 は、CPI_type のテーブルを示す図である。

図 6 7 は、ビデオ EP_map について説明する図である。

図 6 8 は、EP_map について説明する図である。

図 6 9 は、EP_map について説明する図である。

図 7 0 は、EP_map のシンタクスを示す図である。

図 7 1 は、EP_type values のテーブルを示す図である。

図 7 2 は、EP_map_for_one_stream_PID のシンタクスを示す図である。

図 7 3 は、TU_map について説明する図である。

図 7 4 は、TU_map のシンタクスを示す図である。

図 7 5 は、ClipMark のシンタクスを示す図である。

図 7 6 は、mark_type のテーブルを示す図である。

図 7 7 は、mark_type_stamp のテーブルを示す図である。

図 7 8 は、ClipMark のシンタクスの他の例を示す図である。

図 7 9 は、Mark_typeのテーブルの他の例を示す図である。

図 8 0 は、mark_entry()とrepresentative_picture_entry()の例を示す図である。

図 8 1 は、mark_entry()とrepresentative_picture_entry()のシンタクスを示す図である。

図 8 2 は、mark_entry()とrepresentative_picture_entry()のシンタクスの他の例を示す図である。

図 8 3 は、RSPN_ref_EP_startとoffset_num_picturesの関係を説明する図である。

図 8 4 は、mark_entry()とrepresentative_picture_entry()のシンタクスの他の例を示す図である。

図 8 5 は、ClipMarkとEP_mapの関係を説明する図である。

図 8 6 は、menu.thmbとmark.thmbのシンタクスを示す図である。

図 8 7 は、Thumbnailのシンタクスを示す図である。

図 8 8 は、thumbnail_picture_formatのテーブルを示す図である。

図 8 9 A 及び図 8 9 B は、tn_blockについて説明する図である。

図 9 0 は、DVR MPEG 2 のトランスポートストリームの構造について説明する図である。

図 9 1 は、DVR MPEG 2 のトランスポートストリームのレコーダモデルを示す図である。

図 9 2 は、DVR MPEG 2 のトランスポートストリームのプレーヤモデルを示す図である。

図 9 3 は、source packetのシンタクスを示す図である。

図 9 4 は、TP_extra_headerのシンタクスを示す図である。

図 9 5 は、copy permission indicatorのテーブルを示す図である。

図 9 6 は、シームレス接続について説明する図である。

図 9 7 は、シームレス接続について説明する図である。

図 9 8 は、シームレス接続について説明する図である。

図 9 9 は、シームレス接続について説明する図である。

図100は、シームレス接続について説明する図である

図101は、オーディオのオーバーラップについて説明する図である。

図102は、BridgeSequenceを用いたシームレス接続について説明する図である。

図103は、BridgeSequenceを用いないシームレス接続について説明する図である。

図104は、DVR STDモデルを示す図である。

図105は、復号、表示のタイミングチャートを示す図である。

図106は、図81のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

図107は、図81のシンタクスの場合における再生の動作を説明する図である。

図108は、EP_mapの例を示す図である。

図109は、ClipMarkの例を示す図である。

図110は、図81のシンタクスの場合におけるCMスキップ再生処理を説明するフローチャートである。

図111は、図81のシンタクスの場合におけるCMスキップ再生処理を説明するフローチャートである。

図112は、図82のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

図113は、図82のシンタクスの場合における再生を説明する図である。

図114は、EP_mapの例を示す図である。

図115は、ClipMarkの例を示す図である。

図116は、図82のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

図117は、図82のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

図118は、図84のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

図 1 1 9 は、図 8 4 のシンタクスの場合における再生を説明する図である。

図 1 2 0 は、EP_mapの例を示す図である。

図 1 2 1 は、ClipMarkの例を示す図である。

図 1 2 2 は、図 8 4 のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

図 1 2 3 は、図 8 4 のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

図 1 2 4 は、アプリケーションフォーマットを示す図である。

図 1 2 5 は、PlayList上のマークとClip上のマークを説明する図である。

図 1 2 6 は、ClipMarkのシンタクスの他の例を示す図である。

図 1 2 7 は、ClipMarkのシンタクスの更に他の例を示す図である。

図 1 2 8 は、アナログAV信号をエンコードして記録する場合のClipMarkの作成について説明するフローチャートである。

図 1 2 9 は、トランスポートストリームを記録する場合のClipMarkの作成について説明するフローチャートである。

図 1 3 0 は、RealPlayListの作成について説明するフローチャートである。

図 1 3 1 は、VirtualPlayListの作成について説明するフローチャートである。

図 1 3 2 は、PlayListの再生について説明するフローチャートである。

図 1 3 3 は、PlayListMarkの作成について説明するフローチャートである。

図 1 3 4 は、PlayListを再生する際の頭出し再生について説明するフローチャートである。

図 1 3 5 は、PlayListMarkのシンタクスを示す図である。

図 1 3 6 は、PlayListMarkのMark_typeを説明するための図である。

図 1 3 7 は、ClipMarkの他のシンタクスを示す図である。

図 1 3 8 は、ClipMarkのMark_typeを説明するための図である。

図 1 3 9 は、媒体を説明する図である。

発明を実施するための最良の形態

以下に、本発明が適用された実施例について、図面を参照して説明する。図 1 は、本発明を適用した記録再生装置 1 の内部構成例を示す図である。先ず、外部から入力された信号を記録媒体に記録する動作を行う記録部 2 の構成について説明する。記録再生装置 1 は、アナログデータ、又は、デジタルデータを入力し、記録することができる構成とされている。

端子 1 1 には、アナログのビデオ信号が、端子 1 2 には、アナログのオーディオ信号が、それぞれ入力される。端子 1 1 に入力されたビデオ信号は、解析部 1 4 と A V エンコーダ 1 5 に、それぞれ出力される。端子 1 2 に入力されたオーディオ信号は、解析部 1 4 と A V エンコーダ 1 5 に出力される。解析部 1 4 は、入力されたビデオ信号とオーディオ信号からシーンチェンジなどの特徴点を抽出する。

A V エンコーダ 1 5 は、入力されたビデオ信号とオーディオ信号を、それぞれ符号化し、符号化ビデオストリーム (V)、符号化オーディオストリーム (A)、及び A V 同期等のシステム情報 (S) をマルチプレクサ 1 6 に出力する。

符号化ビデオストリームは、例えば、M P E G (Moving Picture Expert Group) 2 方式により符号化されたビデオストリームであり、符号化オーディオストリームは、例えば、M P E G - 1 方式により符号化されたオーディオストリームや、ドルビー A C 3 方式 (商標) により符号化されたオーディオストリーム等である。マルチプレクサ 1 6 は、入力されたビデオ及びオーディオのストリームを、入力システム情報に基づいて多重化して、スイッチ 1 7 を介して多重化ストリーム解析部 1 8 とソースパケッタイザ 1 9 に出力する。

多重化ストリームは、例えば、M P E G - 2 トランスポートストリームや M P E G - 2 プログラムストリームである。ソースパケッタイザ 1 9 は、入力された多重化ストリームを、そのストリームを記録させる記録媒体 1 0 0 のアプリケーションフォーマットに従って、ソースパケットから構成される A V ストリームに符号化する。A V ストリームは、E C C (誤り訂正) 符号化部 2 0 と変調部 2 1 で E C C 符号の付加と変調処理が施され、書込部 2 2 に出力される。書込部 2 2 は、制御部 2 3 から出力される制御信号に基づいて、記録媒体 1 0 0 に A V ストリームファイルを書き込む (記録する)。

デジタルインタフェース又はデジタルテレビジョンチューナから入力されるデジタルテレビジョン放送等のトランスポートストリームは、端子13に入力される。端子13に入力されたトランスポートストリームの記録方式には、2通りあり、これらは、トランスペアレントに記録する方式と、記録ビットレートを下げる等の目的のために再エンコードをした後に記録する方式である。記録方式の指示情報は、ユーザインタフェースとしての端子24から制御部23へ入力される。

入力トランスポートストリームをトランスペアレントに記録する場合、端子13に入力されたトランスポートストリームは、スイッチ17を介して多重化ストリーム解析部18と、ソースパケットタイザ19に出力される。これ以降の記録媒体100へAVストリームが記録されるまでの処理は、上述のアナログの入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

入力トランスポートストリームを再エンコードした後に記録する場合、端子13に入力されたトランスポートストリームは、デマルチプレクサ26に入力される。デマルチプレクサ26は、入力されたトランスポートストリームに対してデマルチプレクス処理を施し、ビデオストリーム(V)、オーディオストリーム(A)、及びシステム情報(S)を抽出する。

デマルチプレクサ26により抽出されたストリーム(情報)の内、ビデオストリームはAVデコーダ27に、オーディオストリームとシステム情報はマルチプレクサ16に、それぞれ出力される。AVデコーダ27は、入力されたビデオストリームを復号し、その再生ビデオ信号をAVエンコーダ15に出力する。AVエンコーダ15は、入力ビデオ信号を符号化し、符号化ビデオストリーム(V)をマルチプレクサ16に出力する。

一方、デマルチプレクサ26から出力され、マルチプレクサ16に入力されたオーディオストリームとシステム情報、及び、AVエンコーダ15から出力されたビデオストリームは、入力システム情報に基づいて、多重化されて、多重化ストリームとして多重化ストリーム解析部18とソースパケットタイザ19にスイッチ17を介して出力される。これ以後の記録媒体100へAVストリームが記

録されるまでの処理は、上述のアナログの入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

以上のような記録再生装置 1 は、A V ストリームのファイルを記録媒体 100 に記録すると共に、そのファイルを説明するアプリケーションデータベース情報も記録する。アプリケーションデータベース情報は、制御部 23 により作成される。制御部 23 への入力情報は、解析部 14 からの動画像の特徴情報、多重化ストリーム解析部 18 からの A V ストリームの特徴情報、及び端子 24 から入力されるユーザからの指示情報である。

解析部 14 から供給される動画像の特徴情報は、A V エンコーダ 15 がビデオ信号を符号化する場合において、解析部 14 により生成されるものである。解析部 14 は、入力ビデオ信号とオーディオ信号の内容を解析し、入力動画像信号の中の特徴的な画像（クリップマーク）に関する情報を生成する。これは、例えば、入力ビデオ信号の中のプログラムの開始点、シーンチェンジ点や C M コマーシャルのスタート点・エンド点、タイトルやテロップなどの特徴的なクリップマーク点の画像の指示情報であり、また、それにはその画像のサムネイルも含まれる。更にオーディオ信号のステレオとモノラルの切り換え点や、無音区間などの情報も含まれる。

これらの画像の指示情報は、制御部 23 を介して、マルチプレクサ 16 へ入力される。マルチプレクサ 16 は、制御部 23 からクリップマークとして指定される符号化ピクチャを多重化する時に、その符号化ピクチャを A V ストリーム上で特定するための情報を制御部 23 に返す。具体的には、この情報は、ピクチャの P T S（プレゼンテーションタイムスタンプ）又はその符号化ピクチャの A V ストリーム上でのアドレス情報である。制御部 23 は、特徴的な画像の種類とその符号化ピクチャを A V ストリーム上で特定するための情報を関連付けて記憶する。

多重化ストリーム解析部 18 からの A V ストリームの特徴情報は、記録される A V ストリームの符号化情報に関する情報であり、解析部 18 により生成される。例えば、A V ストリーム内の I ピクチャのタイムスタンプとアドレス情報、システムタイムクロックの不連続点情報、A V ストリームの符号化パラメータ、A V ストリームの中の符号化パラメータの変化点情報などが含まれる。また、端

子13から入力されるトランスポートストリームをトランスペアレントに記録する場合、多重化ストリーム解析部18は、入力トランスポートストリームの中から前出のクリップマークの画像を検出し、その種類とクリップマークで指定するピクチャを特定するための情報を生成する。

端子24からのユーザの指示情報は、AVストリームの中の、ユーザが指定した再生区間の指定情報、その再生区間の内容を説明するキャラクター文字、ユーザが好みのシーンにセットするブックマークやリジューム点の情報などである。

制御部23は、上記の入力情報に基づいて、AVストリームのデータベース(Clip)、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベース、記録媒体100の記録内容の管理情報(info.dvr)、及びサムネイル画像の情報を作成する。これらの情報から構成されるアプリケーションデータベース情報は、AVストリームと同様にして、ECC符号化部20、変調部21で処理されて、書込部22へ入力される。書込部22は、制御部23から出力される制御信号に基づいて、記録媒体100へデータベースファイルを記録する。

上述したアプリケーションデータベース情報についての詳細は後述する。

このようにして記録媒体100に記録されたAVストリームファイル(画像データと音声データのファイル)と、アプリケーションデータベース情報が再生部3により再生される場合、先ず、制御部23は、読出部28に対して、記録媒体100からアプリケーションデータベース情報を読み出すように指示する。そして、読出部28は、記録媒体100からアプリケーションデータベース情報を読み出し、そのアプリケーションデータベース情報は、復調部29とECC復号部30の復調と誤り訂正処理を経て、制御部23へ入力される。

制御部23は、アプリケーションデータベース情報に基づいて、記録媒体100に記録されているPlayListの一覧を端子24のユーザインタフェースへ出力する。ユーザは、PlayListの一覧から再生したいPlayListを選択し、再生を指定されたPlayListに関する情報が制御部23へ入力される。制御部23は、そのPlayListの再生に必要なAVストリームファイルの読み出しを、読出部28に指示する。読出部28は、その指示に従い、記録媒体100から対応するAVストリームを読み出し復調部29に出力する。復調部29に入力されたAVストリームは、

所定の処理が施されることにより復調され、更にECC復号部30の処理を経て、ソースデパケッタイザ31出力される。

ソースデパケッタイザ31は、記録媒体100から読み出され、所定の処理が施されたアプリケーションフォーマットのAVストリームを、デマルチプレクサ26が処理可能なストリームに変換する。デマルチプレクサ26は、制御部23により指定されたAVストリームの再生区間(PlayItem)を構成するビデオストリーム(V)、オーディオストリーム(A)、及びAV同期等のシステム情報(S)を、AVデコーダ27に出力する。AVデコーダ27は、ビデオストリームとオーディオストリームを復号し、再生ビデオ信号と再生オーディオ信号を、それぞれ対応する端子32と端子33から出力する。

また、ユーザインタフェースとしての端子24から、ランダムアクセス再生や特殊再生を指示する情報が入力された場合、制御部23は、AVストリームのデータベース(Clip)の内容に基づいて、記憶媒体100からのAVストリームの読み出し位置を決定し、そのAVストリームの読み出しを、読出部28に指示する。例えば、ユーザにより選択されたPlayListを、所定の時刻から再生する場合、制御部23は、指定された時刻に最も近いタイムスタンプを持つIピクチャからのデータを読み出すように読出部28に指示する。

また、Clip Informationの中のClipMarkにストアされている番組の頭出し点やシーンチェンジ点の中から、ユーザがあるクリップマークを選択した時(例えば、この動作は、ClipMarkにストアされている番組の頭出し点やシーンチェンジ点のサムネイル画像リストをユーザインタフェースに表示して、ユーザが、その中からある画像を選択することにより行われる)、制御部23は、Clip Informationの内容に基づいて、記録媒体100からのAVストリームの読み出し位置を決定し、そのAVストリームの読み出しを読出部28へ指示する。すなわち、ユーザが選択した画像がストアされているAVストリーム上でのアドレスに最も近いアドレスにあるIピクチャからのデータを読み出すように読出部28へ指示する。読出部28は、指定されたアドレスからデータを読み出し、読み出されたデータは、復調部29、ECC復号部30、ソースデパケッタイザ31の処理を経て、デマルチプレクサ26へ入力され、AVデコーダ27で復号されて、マーク点の

ピクチャのアドレスで示されるAVデータが再生される。

また、ユーザによって高速再生(Fast-forward playback)が指示されたとき、制御部23は、AVストリームのデータベース(Clip)に基づいて、AVストリームの中のIピクチャデータを順次連続して読み出すように読出部28に指示する。

読出部28は、指定されたランダムアクセスポイントからAVストリームのデータを読み出し、読み出されたデータは、後段の各部の処理を経て再生される。

次に、ユーザが、記録媒体100に記録されているAVストリームの編集をするときを説明する。ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合、例えば、番組Aという歌番組から歌手Aの部分を再生し、その後続けて、番組Bという歌番組の歌手Aの部分を再生したいといった再生経路を作成したい場合、ユーザインタフェースとしての端子24から再生区間の開始点(イン点)と終了点(アウト点)の情報が制御部23に入力される。制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成する。

ユーザが、記録媒体100に記録されているAVストリームの一部を消去したいとき、ユーザインタフェースとしての端子24から消去区間のイン点とアウト点の情報が制御部23に入力される。制御部23は、必要なAVストリーム部分だけを参照するようにPlayListのデータベースを変更する。また、AVストリームの不必要なストリーム部分を消去するように、書込部22に指示する。

ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合であり、且つ、それぞれの再生区間をシームレスに接続したい場合について説明する。このようなとき、制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成し、更に、再生区間の接続点付近のビデオストリームの部分的な再エンコードと再多重化を行う。

まず、端子24から再生区間のイン点のピクチャの情報と、アウト点のピクチャの情報が制御部23へ入力される。制御部23は、読出部28にイン点側ピクチャとアウト点側のピクチャを再生するために必要なデータの読み出しを指示する。そして、読出部28は、記録媒体100からデータを読み出し、そのデータ

は、復調部 29、ECC復号部 30、ソースデパケッタイザ 31を経て、デマルチプレクサ 26に出力される。

制御部 23は、デマルチプレクサ 26に入力されたデータを解析して、ビデオストリームの再エンコード方法 (picture_coding_typeの変更、再エンコードする符号化ビット量の割り当て) と、再多重化方式を決定し、その方式をAVエンコーダ 15とマルチプレクサ 16に供給する。

次に、デマルチプレクサ 26は、入力されたストリームをビデオストリーム (V)、オーディオストリーム (A)、及びシステム情報 (S) に分離する。ビデオストリームは、AVデコーダ 27に入力されるデータとマルチプレクサ 16に入力されるデータがある。前者のデータは、再エンコードするために必要なデータであり、これはAVデコーダ 27で復号され、復号されたピクチャはAVエンコーダ 15で再エンコードされて、ビデオストリームにされる。後者のデータは、再エンコードをしないで、オリジナルのストリームからコピーされるデータである。オーディオストリーム、システム情報については、直接、マルチプレクサ 16に入力される。

マルチプレクサ 16は、制御部 23から入力された情報に基づいて、入力ストリームを多重化し、多重化ストリームを出力する。多重化ストリームは、ECC符号化部 20、変調部 21で処理されて、書込部 22に入力される。書込部 22は、制御部 23から供給される制御信号に基づいて、記録媒体 100にAVストリームを記録する。

以下に、アプリケーションデータベース情報や、その情報に基づく再生、編集といった操作に関する説明をする。図 2は、アプリケーションフォーマットの構造を説明する図である。アプリケーションフォーマットは、AVストリームの管理のためにPlayListとClipの2つのレイヤを持つ。Volume Informationは、ディスク内の全てのClipとPlayListの管理をする。ここでは、1つのAVストリームとその付属情報のペアを1つのオブジェクトとし、これをClipという。AVストリームファイルは、Clip AV stream fileといい、その付属情報は、Clip Information fileという。

1つのClip AV stream fileは、MPEG-2トランスポートストリームをアプ

リケーションフォーマットによって規定される構造に配置したデータをストアする。一般的に、ファイルは、バイト列として扱われるが、Clip AV stream fileのコンテンツは、時間軸上に展開され、Clipの中のエントリポイント（Iピクチャ）は、主に時間ベースで指定される。所定のClipへのアクセスポイントのタイムスタンプが与えられた時、Clip Information fileは、Clip AV stream fileの中でデータの読み出しを開始すべきアドレス情報を見つけるために役立つ。

PlayListについて、図3を参照して説明する。PlayListは、Clipの中からユーザが見たい再生区間を選択し、それを簡単に編集することができるようにするために設けられている。1つのPlayListは、Clipの中の再生区間の集まりである。所定のClipの中の1つの再生区間は、PlayItemといい、これは、時間軸上のイン点（I N）とアウト点（O U T）の対で表される。したがって、PlayListは、複数のPlayItemが集まることにより構成される。

PlayListには、2つのタイプがある。1つは、Real PlayListであり、もう1つは、Virtual PlayListである。Real PlayListは、それが参照しているClipのストリーム部分を共有している。すなわち、Real PlayListは、その参照しているClipのストリーム部分に相当するデータ容量をディスクの中で占め、Real PlayListが消去された場合、それが参照しているClipのストリーム部分もまたデータが消去される。

Virtual PlayListは、Clipのデータを共有していない。したがって、Virtual PlayListが変更又は消去されたとしても、Clipの内容には何も変化が生じない。

次に、Real PlayListの編集について説明する。図4 Aは、Real PlayListのクリエイト(create：作成)に関する図であり、A Vストリームが新しいClipとして記録される場合、そのClip全体を参照するReal PlayListが新たに作成される操作である。

図4 Bは、Real PlayListのディバイド(divide：分割)に関する図であり、Real PlayListが所望な点で分けられて、2つのReal PlayListに分割される操作である。この分割という操作は、例えば、1つのPlayListにより管理される1つのクリップ内に、2つの番組が管理されているような場合に、ユーザが1つ1つの番組として登録（記録）し直したいといったようなときに行われる。この操作によ

り、Clipの内容が変更される（Clip自体が分割される）ことはない。

図4 Cは、Real PlayListのコンバイン(combine：結合)に関する図であり、2つのReal PlayListを結合して、1つの新しいReal PlayListにする操作である。この結合という操作は、例えば、ユーザが2つの番組を1つの番組として登録し直したいといったようなときに行われる。この操作により、Clipが変更される（Clip自体が1つにされる）ことはない。

図5 Aは、Real PlayList全体のデリート(delete：削除)に関する図であり、所定のReal PlayList全体を消去する操作がされた場合、削除されたReal PlayListが参照するClipの、対応するストリーム部分も削除される。

図5 Bは、Real PlayListの部分的な削除に関する図であり、Real PlayListの所望な部分が削除されたとき、対応するPlayItemが、必要なClipのストリーム部分だけを参照するように変更される。そして、Clipの対応するストリーム部分は削除される。

図5 Cは、Real PlayListのミニマイズ(Minimize：最小化)に関する図であり、Real PlayListに対応するPlayItemを、Virtual PlayListに必要なClipのストリーム部分だけを参照するようにする操作である。Virtual PlayList にとって unnecessary Clipの、対応するストリーム部分は削除される。

上述したような操作により、Real PlayListが変更されて、そのReal PlayListが参照するClipのストリーム部分が削除された場合、その削除されたClipを使用しているVirtual PlayListが存在し、そのVirtual PlayListにおいて、削除されたClipにより問題が生じる可能性がある。

このようなことが生じないように、ユーザに、削除という操作に対して、「そのReal PlayListが参照しているClipのストリーム部分を参照しているVirtual PlayListが存在し、もし、そのReal PlayListが消去されると、そのVirtual PlayListもまた消去されることになるが、それでも良いか？」といったメッセージなどを表示させることにより、確認（警告）を促した後に、ユーザの指示により削除の処理を実行、又は、キャンセルする。又は、Virtual PlayListを削除する代わりに、Real PlayListに対してミニマイズの操作が行われるようにする。

次に、Virtual PlayListに対する操作について説明する。Virtual PlayListに

対して操作が行われたとしても、Clipの内容が変更されることはない。図6 A及び図6 Bは、アセンブル(Assemble) 編集 (IN-OUT 編集)に関する図であり、ユーザが見たいと所望した再生区間のPlayItemを作り、Virtual PlayListを作成するといった操作である。PlayItem間のシームレス接続が、アプリケーションフォーマットによりサポートされている(後述)。

図6 Aに示したように、2つのReal PlayList 1, 2と、それぞれのReal PlayListに対応するClip1, 2が存在している場合に、ユーザがReal PlayList 1内の所定の区間(In 1乃至Out 1までの区間: PlayItem 1)を再生区間として指示し、続けて再生する区間として、Real PlayList 2内の所定の区間(In 2乃至Out 2までの区間: PlayItem 2)を再生区間として指示したとき、図6 Bに示すように、PlayItem 1とPlayItem 2から構成される1つのVirtual PlayListが作成される。

次に、Virtual PlayListの再編集(Re-editing)について説明する。再編集には、Virtual PlayListの中のイン点やアウト点の変更、Virtual PlayListへの新しいPlayItemの挿入(insert)や追加(append)、Virtual PlayListの中のPlayItemの削除などがある。また、Virtual PlayListそのものを削除することもできる。

図7は、Virtual PlayListへのオーディオのアフレコ(Audio dubbing (post recording))に関する図であり、Virtual PlayListへのオーディオのアフレコをサブパスとして登録する操作のことである。このオーディオのアフレコは、アプリケーションフォーマットによりサポートされている。Virtual PlayListのメインパスのAVストリームに、付加的なオーディオストリームが、サブパスとして付加される。

Real PlayListとVirtual PlayListで共通の操作として、図8に示すようなPlayListの再生順序の変更(Moving)がある。この操作は、ディスク(ボリューム)の中でのPlayListの再生順序の変更であり、アプリケーションフォーマットにおいて定義されるTable Of PlayList (図20などを参照して後述する)によってサポートされる。この操作により、Clipの内容が変更されるようなことはない。

次に、マーク(Mark)について説明する。マークは、図9に示されるように、Clip及びPlayListの中のハイライトや特徴的な時間を指定するために設けられている。Clipに付加されるマークは、ClipMark (クリップマーク)と呼ばれる。Cl

ipMarkは、A Vストリームの内容に起因する特徴的なシーンを指定する、例えば番組の頭だし点やシーンチェンジ点などである。ClipMarkは、図1の例えば解析部14によって生成される。PlayListを再生する時、そのPlayListが参照するClipのマークを参照して、使用することができる。

PlayListに付加されるマークは、PlayListMark（プレイリストマーク）と呼ばれる。PlayListMarkは、主にユーザによってセットされる、例えば、ブックマークやリジューム点などである。Clip又はPlayListにマークをセットすることは、マークの時刻を示すタイムスタンプをマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中から、そのマークのタイムスタンプを除去することである。したがって、マークの設定や削除により、A Vストリームは何の変更もされない。

ClipMarkの別のフォーマットとして、ClipMarkが参照するピクチャをA Vストリームの中でのアドレススペースで指定するようにしてもよい。Clipにマークをセットすることは、マーク点のピクチャを示すアドレススペースの情報をマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中から、そのマーク点のピクチャを示すアドレススペースの情報を除去することである。したがって、マークの設定や削除により、A Vストリームは何の変更もされない。

次に、サムネイルについて説明する。サムネイルは、Volume、PlayList、及びClipに付加される静止画である。サムネイルには、2つの種類があり、1つは、内容を表す代表画としてのサムネイルである。これは主としてユーザがカーソル（図示せず。）などを操作して見たいものを選択するためのメニュー画面で使われるものである。もう1つは、マークが指しているシーンを表す画像である。

Volumeと各Playlistは代表画を持つことができるようにする必要がある。Volumeの代表画は、ディスク（記録媒体100、以下、記録媒体100はディスク状のものであるとし、適宜、ディスクという。）を記録再生装置1の所定の場所にセットした時に、そのディスクの内容を表す静止画を最初に表示する場合などに用いられることを想定している。Playlistの代表画は、Playlistを選択するメニュー画面において、Playlistの内容を表すための静止画として用いられることを

想定している。

Playlistの代表画として、Playlistの最初の画像をサムネイル（代表画）にすることが考えられるが、必ずしも再生時刻0の先頭の画像が内容を表す上で最適な画像とは限らない。そこで、Playlistのサムネイルとして、任意の画像をユーザが設定できるようにする。以上Volumeを表す代表画としてのサムネイルと、Playlistを表す代表画としてのサムネイルの2種類のサムネイルをメニューサムネイルという。メニューサムネイルは頻繁に表示されるため、ディスクから高速に読み出される必要がある。このため、全てのメニューサムネイルを1つのファイルに格納することが効率的である。メニューサムネイルは、必ずしもボリューム内の動画から抜き出したピクチャである必要はなく、図10に示すように、パーソナルコンピュータやデジタルスチルカメラから取り込まれた画像でもよい。

一方、ClipとPlaylistには、複数個のマークを打てる必要があり、マーク位置の内容を知るためにマーク点の画像を容易に見ることが出来るようにする必要がある。このようなマーク点を表すピクチャをマークサムネイル（Mark Thumbnail）という。したがって、マークサムネイルの元となる画像は、外部から取り込んだ画像よりも、マーク点の画像を抜き出したものが主となる。

図11は、Playlistに付けられるマークと、そのマークサムネイルの関係について示す図であり、図12は、Clipに付けられるマークと、そのマークサムネイルの関係について示す図である。マークサムネイルは、メニューサムネイルと異なり、Playlistの詳細を表す時に、サブメニュー等で使われるため、短いアクセス時間で読み出されるようなことは要求されない。そのため、サムネイルが必要になる度に、記録再生装置1がファイルを開き、そのファイルの一部を読み出すことで多少時間がかかっても、問題にはならない。

また、ボリューム内に存在するファイル数を減らすために、全てのマークサムネイルは1つのファイルに格納するのがよい。Playlistはメニューサムネイル1つと複数のマークサムネイルを有することができるが、Clipは直接ユーザが選択する必要性がない（通常、Playlist経由で指定する）ため、メニューサムネイルを設ける必要はない。

図13は、上述したことを考慮した場合のメニューサムネイル、マークサムネ

イル、PlayList、及びClipの関係について示した図である。メニューサムネイルファイルには、PlayList毎に設けられたメニューサムネイルがファイルされている。メニューサムネイルファイルには、ディスクに記録されているデータの内容を代表するボリュームサムネイルが含まれている。マークサムネイルファイルは、各PlayList毎と各Clip毎に作成されたサムネイルがファイルされている。

次に、CPI (Characteristic Point Information) について説明する。CPIは、Clipインフォメーションファイルに含まれるデータであり、主に、それはClipへのアクセスポイントのタイムスタンプが与えられた時、Clip AV stream fileの中でデータの読み出しを開始すべきデータアドレスを見つけるために用いられる。本例では、2種類のCPIを用いる。1つは、EP_mapであり、もう一つは、TU_mapである。

EP_mapは、エントリポイント (EP) データのリストであり、それはエレメンタリーストリーム及びトランスポートストリームから抽出されたものである。これは、AVストリームの中でデコードを開始すべきエントリポイントの場所を見つけるためのアドレス情報を持つ。1つのEPデータは、プレゼンテーションタイムスタンプ (PTS) と、このPTSに対応するアクセスユニットのAVストリームの中のデータアドレスの対で構成される。

EP_mapは、主に2つの目的のために使用される。第1に、PlayListの中でプレゼンテーションタイムスタンプによって参照されるアクセスユニットのAVストリームの中のデータアドレスを見つけるために使用される。第2に、ファーストフォワード再生やファーストリバース再生のために使用される。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができる時、EP_mapが作成され、ディスクに記録される。

TU_mapは、デジタルインタフェースを通して入力されるトランスポートパケットの到着時刻に基づいたタイムユニット (TU) データのリストを持つ。これは、到着時刻ベースの時間とAVストリームの中のデータアドレスとの関係を与える。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができない時、TU_mapが作成され、ディスクに記録される。

STCInfoは、MPEG-2トランスポートストリームをストアしているAVストリームファイルの中にあるSTCの不連続点情報をストアする。仮に、AVストリームがSTCの不連続点を持つ場合、そのAVストリームファイルの中で同じ値のPTSが現れる可能性がある。このため、AVストリーム上の所定の時刻をPTSベースで指すとき、アクセスポイントのPTSだけではそのポイントを特定するためには不十分である。

更に、このPTSを含むところの連続なSTC区間のインデックスが必要である。連続なSTC区間を、このフォーマットでは、STC-sequenceといい、そのインデックスをSTC-sequence-idという。STC-sequenceの情報は、Clip Information fileのSTCInfoで定義される。STC-sequence-idは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリームファイルではオプションである。

プログラムは、エレメンタリストリームの集まりであり、これらのストリームの同期再生のために、ただ1つのシステムタイムベースを共有するものである。再生装置にとって、AVストリームのデコードに先だち、そのAVストリームの内容が分かることは有用である。例えば、ビデオやオーディオのエレメンタリストリームを伝送するトランスポートパケットのPIDの値や、ビデオやオーディオのコンポーネント種類（例えば、HDTVのビデオとMPEG-2 AACのオーディオストリームなど）などの情報である。

この情報はAVストリームを参照するところのPlayListの内容をユーザに説明するところのメニュー画面を作成するのに有用であるし、また、AVストリームのデコードに先だって、再生装置のAVデコーダ及びデマルチプレクサの初期状態をセットするために役立つ。この理由のために、Clip Information fileは、プログラムの内容を説明するためのProgramInfoを持つ。

MPEG-2トランスポートストリームをストアしているAVストリームファイルは、ファイルの中でプログラム内容が変化するかもしれない。例えば、ビデオエレメンタリストリームを伝送するところのトランスポートパケットのPIDが変化したり、ビデオストリームのコンポーネント種類がSDTVからHDTVに変化する等である。

ProgramInfoは、AVストリームファイルの中でのプログラム内容の変化点の情報をストアする。AVストリームファイルの中で、このフォーマットで定めるところのプログラム内容が一定である区間をProgram-sequenceという。Program-sequenceは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリームファイルではオプションである。

本例では、セルフエンコードのストリームフォーマット (SESF) を定義する。SESFは、アナログ入力信号を符号化する目的、及びデジタル入力信号 (例えばDV) をデコードしてからMP E G-2トランスポートストリームに符号化する場合に用いられる。

SESFは、MP E G-2トランスポートストリーム及びAVストリームについてのエレメンタリーストリームの符号化制限を定義する。記録再生装置1が、SESFストリームをエンコードし、記録する場合、EP_mapが作成され、ディスクに記録される。

デジタル放送のストリームは、次に示す方式の内のいずれかが用いられて記録媒体100に記録される。まず、デジタル放送のストリームをSESFストリームにトランスコーディングする。この場合、記録されたストリームは、SESFに準拠しなければならない。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

或いは、デジタル放送ストリームを構成するエレメンタリーストリームを新しいエレメンタリーストリームにトランスコーディングし、そのデジタル放送ストリームの規格化組織が定めるストリームフォーマットに準拠した新しいトランスポートストリームに再多重化する。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

例えば、入力ストリームがISDB (日本のデジタルBS放送の規格名称) 準拠のMP E G-2トランスポートストリームであり、それがHDTVビデオストリームとMP E G AACオーディオストリームを含むとする。HDTVビデオストリームをSDTVビデオストリームにトランスコーディングし、このSDTVビデオストリームとオリジナルのAACオーディオストリームをTSに再多重化する。SDTVストリームと記録されるトランスポートストリームは、共にI

SDBフォーマットに準拠しなければならない。

デジタル放送のストリームが、記録媒体100に記録される際の他の方式として、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、その時にEP_mapが作成されてディスクに記録される。

又は、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、その時にTU_mapが作成されてディスクに記録される。

次に、ディレクトリとファイルについて説明する。以下、記録再生装置1をDVR (Digital Video Recording) と適宜記述する。図14はディスク上のディレクトリ構造の一例を示す図である。DVRのディスク上に必要なディレクトリは、図14に示したように、“DVR”ディレクトリを含むrootディレクトリ、“PLAYLIST”ディレクトリ、“CLIPINF”ディレクトリ、“M2TS”ディレクトリ、及び“DATA”ディレクトリを含む“DVR”ディレクトリである。rootディレクトリの下に、これら以外のディレクトリを作成されるようにしてもよいが、これらは、ここでのアプリケーションフォーマットでは、無視されとする。

“DVR”ディレクトリの下には、DVRアプリケーションフォーマットによって規定される全てのファイルとディレクトリがストアされる。“DVR”ディレクトリは、4個のディレクトリを含む。“PLAYLIST”ディレクトリの下には、Real PlayListとVirtual PlayListのデータベースファイルが置かれる。このディレクトリは、PlayListが1つもなくても存在する。

“CLIPINF”ディレクトリの下には、Clipのデータベースが置かれる。このディレクトリも、Clipが1つもなくても存在する。“M2TS”ディレクトリの下には、AVストリームファイルが置かれる。このディレクトリは、AVストリームファイルが1つもなくても存在する。“DATA”ディレクトリは、デジタルTV放送などのデータ放送のファイルがストアされる。

“DVR”ディレクトリは、次に示すファイルをストアする。“info.dvr”ファイルは、“DVR”ディレクトリの下に作られ、アプリケーションレイヤの全体的な情報をストアする。“DVR”ディレクトリの下には、ただ1つのinfo.dvrがなければならない。

ファイル名は、info.dvrに固定されれるとする。"menu.thmb"ファイルは、メニューサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、0又は1つのメニューサムネイルがなければならない。ファイル名は、menu.thmbに固定されれるとする。メニューサムネイル画像が1つもない場合、このファイルは、存在しなくてもよい。

"mark.thmb"ファイルは、マークサムネイル画像に関連する情報をストアする。"DVR"ディレクトリの下には、0又は1つのマークサムネイルがなければならない。ファイル名は、mark.thmbに固定されれるとする。メニューサムネイル画像が1つもない場合、このファイルは、存在しなくてもよい。

"PLAYLIST"ディレクトリは、2種類のPlayListファイルをストアするものであり、これらは、Real PlayListとVirtual PlayListである。"xxxxx.rpls"ファイルは、1つのReal PlayListに関連する情報をストアする。それぞれのReal PlayList毎に、1つのファイルが作られる。ファイル名は、"xxxxx.rpls"である。ここで、"xxxxx"は、5個の0乃至9まで数字である。ファイル拡張子は、"rpls"でなければならないとする。

"yyyyy.vpls"ファイルは、1つのVirtual PlayListに関連する情報をストアする。それぞれのVirtual PlayList毎に、1つのファイルが作られる。ファイル名は、"yyyyy.vpls"である。ここで、"yyyyy"は、5個の0乃至9まで数字である。ファイル拡張子は、"vpls"でなければならないとする。

"CLIPINF"ディレクトリは、それぞれのAVストリームファイルに対応して、1つのファイルをストアする。"zzzzz.clpi"ファイルは、1つのAVストリームファイル(Clip AV stream file 又は Bridge-Clip AV stream file)に対応するClip Information fileである。ファイル名は、"zzzzz.clpi"であり、"zzzzz"は、5個の0乃至9までの数字である。ファイル拡張子は、"clpi"でなければならないとする。

"M2TS"ディレクトリは、AVストリームのファイルをストアする。"zzzzz.m2ts"ファイルは、DVRシステムにより扱われるAVストリームファイルである。これは、Clip AV stream file又はBridge-Clip AV streamである。ファイル名は、"zzzzz.m2ts"であり、"zzzzz"は、5個の0乃至9までの数字である。ファイル拡張

張子は、"m2ts"でなければならないとする。

"DATA" ディレクトリは、データ放送から伝送されるデータをストアするものであり、データとは、例えば、XML fileやMHEGファイルなどである。

次に、各ディレクトリ（ファイル）のシンタクスとセマンティクスを説明する。先ず、"info.dvr" ファイルについて説明する。図15は、"info.dvr" ファイルのシンタクスを示す図である。"info.dvr" ファイルは、3個のオブジェクトから構成され、これらは、DVRVolume()、TableOfPlayLists()、及びMakersPrivateData()である。

図15に示したinfo.dvrのシンタクスについて説明するに、TableOfPlayLists_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、TableOfPlayList()の先頭アドレスを示す。相対バイト数は0からカウントされる。

MakersPrivateData_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数は0からカウントされる。padding_word（パディングワード）は、info.dvrのシンタクスに従って挿入される。N1とN2は、0又は任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしてもよい。

DVRVolume()は、ボリューム（ディスク）の内容を記述する情報をストアする。図16は、DVRVolume()のシンタクスを示す図である。図16に示したDVRVolume()のシンタクスを説明するに、version_numberは、このDVRVolume()のバージョンナンバを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、"0045"と符号化される。

lengthは、このlengthフィールドの直後からDVRVolume()の最後までのDVRVolume()のバイト数を示す32ビットの符号なし整数で表される。

ResumeVolume()は、ボリュームの中で最後に再生したReal PlayList又はVirtual PlayListのファイル名を記憶している。但し、Real PlayList又はVirtual PlayListの再生をユーザが中断した時の再生位置は、PlayListMark()において定義されるresume-markにストアされる（図42、図43）。

図17は、ResumeVolume()のシンタクスを示す図である。図17に示したResu

meVolume()のシンタクスを説明するに、valid_flagは、この1ビットのフラグが1にセットされている場合、resume_PlayList_nameフィールドが有効であることを示し、このフラグが0にセットされている場合、resume_PlayList_nameフィールドが無効であることを示す。

resume_PlayList_nameの10バイトのフィールドは、リジュームされるべきReal PlayList又はVirtual PlayListのファイル名を示す。

図16に示したDVRVolume()のシンタクスの中の、UIAppInfoVolume は、ボリュームについてのユーザインタフェースアプリケーションのパラメータをストアする。図18は、UIAppInfoVolumeのシンタクスを示す図であり、そのセマンティクスを説明するに、character_setの8ビットのフィールドは、Volume_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示される値に対応する。

name_lengthの8ビットフィールドは、Volume_nameフィールドの中に示されるボリューム名のバイト長を示す。Volume_nameのフィールドは、ボリュームの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、これはボリュームの名称を示す。Volume_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていてもよい。

Volume_protect_flagは、ボリュームの中のコンテンツを、ユーザに制限することなしに見せてよいかどうかを示すフラグである。このフラグが1にセットされている場合、ユーザが正しくPIN番号（パスワード）を入力できたときだけ、そのボリュームのコンテンツを、ユーザに見せること（再生されること）が許可される。このフラグが0にセットされているとき、ユーザがPIN番号を入力しなくても、そのボリュームのコンテンツを、ユーザに見せることが許可される。

最初に、ユーザが、ディスクをプレーヤへ挿入した時点において、もしこのフラグが0にセットされているか、又は、このフラグが1にセットされていてもユーザがPIN番号を正しく入力できたならば、記録再生装置1は、そのディスクの中のPlayListの一覧を表示させる。それぞれのPlayListの再生制限は、volume_protect_flagとは無関係であり、それはUIAppInfoPlayList()の中に定義されるplayback_control_flagによって示される。

P I Nは、4個の0乃至9までの数字で構成され、それぞれの数字は、I S O / I E C 6 4 6に従って符号化される。ref_thumbnail_indexのフィールドは、ボリュームに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのボリュームにはサムネイル画像が付加されており、そのサムネイル画像は、menu.thumファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのボリュームにはサムネイル画像が付加されていないことを示す。

次に、図 1 5 に示したinfo.dvrのシンタクス内のTableOfPlayLists()について説明する。TableOfPlayLists()は、PlayList(Real PlayListとVirtual PlayList)のファイル名をストアする。ボリュームに記録されている全てのPlayListファイルは、TableOfPlayList()の中に含まれる。TableOfPlayLists()は、ボリュームの中のPlayListのデフォルトの再生順序を示す。

図 2 0 は、TableOfPlayLists()のシンタクスを示す図であり、そのシンタクスについて説明するに、TableOfPlayListsのversion_numberは、このTableOfPlayListsのバージョンナンバを示す4個のキャラクター文字を示す。version_numberは、I S O 6 4 6に従って、"0045"と符号化されなければならない。

lengthは、このlengthフィールドの直後からTableOfPlayLists()の最後までのTableOfPlayLists()のバイト数を示す32ビットの符号なしの整数である。number_of_PlayListsの16ビットのフィールドは、PlayList_file_nameを含むfor-loopのループ回数を示す。この数字は、ボリュームに記録されているPlayListの数に等しくなければならない。PlayList_file_nameの10バイトの数字は、PlayListのファイル名を示す。

図 2 1 は、TableOfPlayLists()のシンタクスの別の構成を示す図である。図 2 1 に示したシンタクスは、図 2 0 に示したシンタクスに、UIAppinfoPlayList (後述)を含ませた構成とされている。このように、UIAppinfoPlayListを含ませた構成とすることで、TableOfPlayListsを読み出すだけで、メニュー画面を作成することが可能となる。ここでは、図 2 0 に示したシンタクスを用いるとして以下の説明をする。

図 1 5 に示した info.dvr のシンタクス内の MakersPrivateData について説明する。MakersPrivateData は、記録再生装置 1 のメーカーが、各社の特別なアプリケーションのために、MakersPrivateData() の中にメーカーのプライベートデータを挿入できるように設けられている。各メーカーのプライベートデータは、それを定義したメーカーを識別するために標準化された maker_ID を持つ。MakersPrivateData() は、1 つ以上の maker_ID を含んでもよい。

所定のメーカーが、プライベートデータを挿入したい時に、既に他のメーカーのプライベートデータが MakersPrivateData() に含まれていた場合、他のメーカーは、既にある古いプライベートデータを消去するのではなく、新しいプライベートデータを MakersPrivateData() の中に追加するようにする。このように、ここでは、複数のメーカーのプライベートデータが、1 つの MakersPrivateData() に含まれることが可能であるようにする。

図 2 2 は、MakersPrivateData のシンタクスを示す図である。図 2 2 に示した MakersPrivateData のシンタクスについて説明するに、version_number は、この MakersPrivateData() のバージョンナンバを示す 4 個のキャラクター文字を示す。version_number は、ISO 646 に従って、"0045" と符号化されなければならない。length は、この length フィールドの直後から MakersPrivateData() の最後まで MakersPrivateData() のバイト数を示す 32 ビットの符号なし整数を示す。

mpd_blocks_start_address は、MakersPrivateData() の先頭のバイトからの相対バイト数を単位として、最初の mpd_block() の先頭バイトアドレスを示す。相対バイト数は 0 からカウントされる。number_of_maker_entries は、MakersPrivateData() の中に含まれているメーカープライベートデータのエン트리数を与える 16 ビットの符号なし整数である。MakersPrivateData() の中に、同じ maker_ID の値を持つメーカープライベートデータが 2 個以上存在してはならない。

mpd_block_size は、1024 バイトを単位として、1 つの mpd_block の大きさを与える 16 ビットの符号なし整数である。例えば、mpd_block_size=1 ならば、これは 1 つの mpd_block の大きさが 1024 バイトであることを示す。number_of_mpd_blocks は、MakersPrivateData() の中に含まれる mpd_block の数を与える 16 ビットの符号なし整数である。maker_ID は、そのメーカープライベートデータを作成

したDVRシステムの製造メーカを示す16ビットの符号なし整数である。maker_IDに符号化される値は、このDVRフォーマットのライセンスによって指定される。

maker_model_codeは、そのメーカプライベートデータを作成したDVRシステムのモデルナンバコードを示す16ビットの符号なし整数である。maker_model_codeに符号化される値は、このフォーマットのライセンスを受けた製造メーカによって設定される。start_mpd_block_numberは、そのメーカプライベートデータが開始されるmpd_blockの番号を示す16ビットの符号なし整数である。メーカプライベートデータの先頭データは、mpd_blockの先頭にアラインされなければならない。start_mpd_block_numberは、mpd_blockのfor-loopの中の変数jに対応する。

mpd_lengthは、バイト単位でメーカプライベートデータの大きさを示す32ビットの符号なし整数である。mpd_blockは、メーカプライベートデータがストアされる領域である。MakersPrivateData()の中の全てのmpd_blockは、同じサイズでなければならない。

次に、Real PlayList fileとVirtual PlayList fileについて、換言すれば、xxxx.rplsとyyyy.vplsについて説明する。図23は、xxxx.rpls (Real PlayList)、又は、yyyy.vpls (Virtual PlayList) のシンタクスを示す図である。xxx.rplsとyyyy.vplsは、同一のシンタクス構成を持つ。xxxx.rplsとyyyy.vplsは、それぞれ、3個のオブジェクトから構成され、これらは、PlayList()、PlayListMark()、及びMakersPrivateData()である。

PlayListMark_Start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、PlayListMark()の先頭アドレスを示す。相対バイト数は0からカウントされる。

MakersPrivateData_Start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数は0からカウントされる。

padding_word (パディングワード) は、PlayListファイルのシンタクスに従って挿入され、N1とN2は、0又は任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしてもよい。

ここで、既に、簡便に説明したが、PlayListについて更に説明する。ディスク内にある全てのReal PlayListによって、Bridge-Clip（後述）を除く全てのClipの中の再生区間が参照されていなければならない。且つ、2つ以上のReal PlayListが、それらのPlayItemで示される再生区間を同一のClipの中でオーバーラップさせてはならない。

図24A～図24Cを参照して更に説明するに、図24Aに示したように、全てのClipは、対応するReal PlayListが存在する。この規則は、図24Bに示したように、編集作業が行われた後においても守られる。したがって、全てのClipは、どれかしらのReal PlayListを参照することにより、必ず視聴することが可能である。

図24Cに示したように、Virtual PlayListの再生区間は、Real PlayListの再生区間又はBridge-Clipの再生区間の中に含まれていなければならない。どのVirtual PlayListにも参照されないBridge-Clipがディスクの中に存在してはならない。

Real PlayListは、PlayItemのリストを含むが、SubPlayItemを含んではならない。Virtual PlayListは、PlayItemのリストを含み、PlayList()の中に示されるCPI_typeがEP_map typeであり、且つPlayList_typeが0（ビデオとオーディオを含むPlayList）である場合、Virtual PlayListは、1つのSubPlayItemを含むことができる。本実施例におけるPlayList()では、SubPlayItemはオーディオのアフレコの目的にだけに使用される、そして、1つのVirtual PlayListが持つSubPlayItemの数は、0又は1でなければならない。

次に、PlayListについて説明する。図25は、PlayListのシンタクスを示す図である。図25に示したPlayListのシンタクスを説明するに、version_numberは、このPlayList()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からPlayList()の最後までPlayList()のバイト数を示す32ビットの符号なし整数である。PlayList_typeは、このPlayListのタイプを示す8ビットのフィールドであり、その一例を図26に示す。

CPI_typeは、1ビットのフラグであり、PlayItem()及びSubPlayItem()によって

参照されるClipのCPI_typeの値を示す。1つのPlayListによって参照される全てのClipは、これらのCPI()の中に定義されるCPI_typeの値が同じでなければならない。number_of_PlayItemsは、PlayListの中にあるPlayItemの数を示す16ビットのフィールドである。

所定のPlayItem()に対応するPlayItem_idは、PlayItem()を含むfor-loopの中で、そのPlayItem()の現れる順番により定義される。PlayItem_idは、0から開始される。number_of_SubPlayItemsは、PlayListの中にあるSubPlayItemの数を示す16ビットのフィールドである。この値は、0又は1である。付加的なオーディオストリームパス（オーディオストリームパス）は、サブパスの一種である。

次に、図25に示したPlayListのシンタクスのUIAppInfoPlayListについて説明する。UIAppInfoPlayListは、PlayListについてのユーザインタフェースアプリケーションのパラメータをストアする。図27は、UIAppInfoPlayListのシンタクスを示す図である。図27に示したUIAppInfoPlayListのシンタクスを説明するに、character_setは、8ビットのフィールドであり、PlayList_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示したテーブルに準拠する値に対応する。

name_lengthは、8ビットフィールドであり、PlayList_nameフィールドの中に示されるPlayList名のバイト長を示す。PlayList_nameのフィールドは、PlayListの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはPlayListの名称を示す。PlayList_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていてもよい。

record_time_and_dateは、PlayListが記録された時の日時をストアする56ビットのフィールドである。このフィールドは、年/月/日/時/分/秒について、14個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、2001/12/23:01:02:03 は、“0x20011223010203”と符号化される。

durationは、PlayListの総再生時間を時間/分/秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、01:45:30は、“0x014530”

と符号化される。

valid_periodは、PlayListが有効である期間を示す32ビットのフィールドである。このフィールドは、8個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、記録再生装置1は、この有効期間の過ぎたPlayListを自動消去する、といったように用いられる。例えば、2001/05/07 は、"0x20010507"と符号化される。

maker_idは、そのPlayListを最後に更新したDVRプレーヤ（記録再生装置1）の製造者を示す16ビットの符号なし整数である。maker_idに符号化される値は、DVRフォーマットのライセンサによって割り当てられる。maker_codeは、そのPlayListを最後に更新したDVRプレーヤのモデル番号を示す16ビットの符号なし整数である。maker_codeに符号化される値は、DVRフォーマットのライセンスを受けた製造者によって決められる。

playback_control_flagのフラグが1にセットされている場合、ユーザが正しくPIN番号を入力できた場合にだけ、そのPlayListは再生される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、ユーザは、このPlayListを視聴することができる。

write_protect_flagは、図28Aにテーブルを示すように、1にセットされている場合、write_protect_flagを除いて、そのPlayListの内容は、消去及び変更されない。このフラグが0にセットされている場合、ユーザは、そのPlayListを自由に消去及び変更できる。このフラグが1にセットされている場合、ユーザが、そのPlayListを消去、編集、又は上書きする前に、記録再生装置1はユーザに再確認するようなメッセージを表示させる。

write_protect_flagが0にセットされているReal PlayListが存在し、且つ、そのReal PlayListのClipを参照するVirtual PlayListが存在し、そのVirtual PlayListのwrite_protect_flagが1にセットされていてもよい。ユーザが、Real PlayListを消去しようとする場合、記録再生装置1は、そのReal PlayListを消去する前に、上記Virtual PlayListの存在をユーザに警告するか、又は、このReal PlayListを"Minimize" する。

is_played_flagは、図28Bに示すように、フラグが1にセットされている場

合、そのPlayListは、記録されてから一度は再生されたことを示し、0にセットされている場合、このPlayListは、記録されてから一度も再生されたことがないことを示す。

archiveは、図28Cに示すように、そのPlayListがオリジナルであるか、コピーされたものであるかを示す2ビットのフィールドである。ref_thumbnail_indexのフィールドは、PlayListを代表するサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのPlayListには、PlayListを代表するサムネイル画像が付加されており、そのサムネイル画像は、menu.thumファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFであるとき、そのPlayListには、PlayListを代表するサムネイル画像が付加されていない。

次に、PlayItemについて説明する。1つのPlayItem()は、基本的に次のデータを含む。Clipのファイル名を指定するためのClip_information_file_name、Clipの再生区間を特定するためのIN_timeとOUT_timeのペア、PlayList()において定義されるCPI_typeがEP_map typeである場合、IN_timeとOUT_timeが参照するところのSTC_sequence_id、及び、先行するPlayItemと現在のPlayItemとの接続の状態を示すところのconnection_conditionである。

PlayListが2つ以上のPlayItemから構成される時、それらのPlayItemはPlayListのグローバル時間軸上に、時間のギャップ又はオーバーラップなしに一系列に並べられる。PlayList()において定義されるCPI_typeがEP_map typeであり、且つ現在のPlayItemがBridgeSequence()を持たない時、そのPlayItemにおいて定義されるIN_timeとOUT_timeのペアは、STC_sequence_idによって指定される同じSTC連続区間上の時間を指していなければならない。そのような例を図29に示す。

図30は、PlayList()において定義されるCPI_typeがEP_map typeであり、且つ現在のPlayItemがBridgeSequence()を持つ時、次に説明する規則が適用される場合を示している。現在のPlayItemに先行するPlayItemのIN_time (図の中でIN_time1と示されているもの)は、先行するPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を指している。先行するPlayItemのOUT_time (図の

中でOUT_time1と示されているもの)は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このOUT_timeは、後述する符号化制限に従っていなければならない。

現在のPlayItemのIN_time (図の中でIN_time2と示されているもの)は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このIN_timeも、後述する符号化制限に従っていなければならない。現在のPlayItemのOUT_time (図の中でOUT_time2と示されているもの)は、現在のPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を指している。

図31に示すように、PlayList()のCPI_typeがTU_map typeである場合、PlayItemのIN_timeとOUT_timeのペアは、同じClip AVストリーム上の時間を指している。

PlayItemのシンタクスは、図32に示すようになる。図32に示したPlayItemのシンタクスを説明するに、Clip_Information_file_nameのフィールドは、Clip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

STC_sequence_idは、8ビットのフィールドであり、PlayItemが参照するSTC連続区間のSTC_sequence_idを示す。PlayList()の中で指定されるCPI_typeがTU_map typeである場合、この8ビットフィールドは何も意味を持たず、0にセットされる。IN_timeは、32ビットフィールドであり、PlayItemの再生開始時刻をストアする。IN_timeのセマンティクスは、図33に示すように、PlayList()において定義されるCPI_typeによって異なる。

OUT_timeは、32ビットフィールドであり、PlayItemの再生終了時刻をストアする。OUT_timeのセマンティクスは、図34に示すように、PlayList()において定義されるCPI_typeによって異なる。

Connection_Conditionは、図35に示したような先行するPlayItemと、現在のPlayItemとの間の接続状態を示す2ビットのフィールドである。図36A～図36Dは、図35に示したConnection_Conditionの各状態について説明する図であ

る。

次に、BridgeSequenceInfoについて、図37を参照して説明する。BridgeSequenceInfo()は、現在のPlayItemの付属情報であり、次に示す情報を持つ。Bridge-Clip AV streamファイルとそれに対応するClip Information file (図45)を指定するBridge_Clip_Information_file_nameを含む。

また、先行するPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。このアドレスは、RSPN_exit_from_previous_Clipと称される。更に現在のPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。このアドレスは、RSPN_enter_to_current_Clipという。

図37において、RSPN_arrival_time_discontinuityは、the Bridge-Clip AV streamファイルの中でアライバルタイムベースの不連続点があるところのソースパケットのアドレスを示す。このアドレスは、ClipInfo() (図46) の中において定義される。

図38は、BridgeSequenceinfoのシンタクスを示す図である。図38に示したBridgeSequenceinfoのシンタクスを説明するに、Bridge_Clip_Information_file_nameのフィールドは、Bridge-Clip AV streamファイルに対応するClip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、'Bridge-Clip AV stream'を示していなければならない。

RSPN_exit_from_previous_Clipの32ビットフィールドは、先行するPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、先行するPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

RSPN_enter_to_current_Clipの32ビットフィールドは、現在のPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、現在のPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

次に、SubPlayItemについて、図39を参照して説明する。SubPlayItem()の使用は、Playlist()のCPI_typeがEP_map typeである場合に許される。ここでは、SubPlayItemはオーディオのアフレコの目的のためだけに使用されとする。SubPlayItem()は、次に示すデータを含む。まず、Playlistの中のsub pathが参照するClipを指定するためのClip_information_file_nameを含む。

また、Clipの中のsub pathの再生区間を指定するためのSubPath_IN_time と SubPath_OUT_timeを含む。更に、main pathの時間軸上でsub pathが再生開始する時刻を指定するためのsync_PlayItem_id と sync_start_PTS_of_PlayItemを含む。sub pathに参照されるオーディオのClip AV streamは、S T C不連続点（システムタイムベースの不連続点）を含んではならない。sub pathに使われるClipのオーディオサンプルのクロックは、main pathのオーディオサンプルのクロックにロックされている。

図40は、SubPlayItemのシンタクスを示す図である。図40に示したSubPlayItemのシンタクスを説明するに、Clip_Information_file_nameのフィールドは、Clip Information fileのファイル名を示し、これはPlaylistの中でsub pathによって使用される。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

SubPath_typeの8ビットのフィールドは、sub pathのタイプを示す。ここでは、図41に示すように、'0x00'しか設定されておらず、他の値は、将来のために確保されている。

sync_PlayItem_idの8ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻が含まれるPlayItemのPlayItem_idを示す。所定のPlayItem

に対応するPlayItem_idの値は、PlayList()において定義される（図25参照）。

sync_start_PTS_of_PlayItemの32ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻を示し、sync_PlayItem_idで参照されるPlayItem上のPTS(Presentation Time Stamp)の上位32ビットを示す。SubPath_IN_timeの32ビットフィールドは、Sub pathの再生開始時刻をストアする。SubPath_IN_timeは、Sub Pathの中で最初のプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示す。

SubPath_OUT_timeの32ビットフィールドは、Sub pathの再生終了時刻をストアする。SubPath_OUT_timeは、次式によって算出されるPresentation_end_TSの値の上位32ビットを示す。

$$\text{Presentation_end_TS} = \text{PTS_out} + \text{AU_duration}$$

ここで、PTS_outは、SubPathの最後のプレゼンテーションユニットに対応する33ビット長のPTSである。AU_durationは、SubPathの最後のプレゼンテーションユニットの90kHz単位の表示期間である。

次に、図23に示したxxxxx.rplsとyyyyy.vplsのシンタクス内のPlayListMark()について説明する。PlayListについてのマーク情報は、このPlayListMarkにストアされる。図42は、PlayListMarkのシンタクスを示す図である。図42に示したPlayListMarkのシンタクスについて説明するに、version_numberは、このPlayListMark()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からPlayListMark()の最後までのPlayListMark()のバイト数を示す32ビットの符号なし整数である。number_of_PlayList_marksは、PlayListMarkの中にストアされているマークの個数を示す16ビットの符号なし整数である。number_of_PlayList_marks は、0であってもよい。mark_typeは、マークのタイプを示す8ビットのフィールドであり、図43に示すテーブルに従って符号化される。

mark_time_stampの32ビットフィールドは、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図44に示すように、PlayList()において定義されるCPI_typeによって異なる。PlayItem

_idは、マークが置かれているところのPlayItemを指定する8ビットのフィールドである。所定のPlayItemに対応するPlayItem_idの値は、PlayList()において定義される(図25参照)。

character_setの8ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示した値に対応する。name_lengthの8ビットフィールドは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、これはマークの名称を示す。Mark_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どのような値が設定されてもよい。

ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される(後述)。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのマークにはサムネイル画像が付加されていないことを示す。

次に、Clip information fileについて説明する。zzzzz.clpi (Clip information fileファイル)は、図45に示すように6個のオブジェクトから構成される。これらは、ClipInfo()、STC_Info()、ProgramInfo()、CPI()、ClipMark()、及びMakersPrivateData()である。AVストリーム(Clip AVストリーム又はBridge-Clip AV stream)とそれに対応するClip Informationファイルは、同じ数字列の”zzzzz”が使用される。

図45に示したzzzzz.clpi (Clip information fileファイル)のシンタクスについて説明するに、ClipInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipInfo()の先頭アドレスを示す。相対バイト数は0からカウントされる。

STC_Info_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、STC_Info()の先頭アドレスを示す。相対バイト数は0から

カウントされる。ProgramInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ProgramInfo()の先頭アドレスを示す。相対バイト数は0からカウントされる。CPI_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、CPI()の先頭アドレスを示す。相対バイト数は0からカウントされる。

ClipMark_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipMark()の先頭アドレスを示す。相対バイト数は0からカウントされる。MakersPrivateData_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数は0からカウントされる。padding_word (パディングワード) は、zzzzz.clpiファイルのシンタクスにしたがって挿入される。N1, N2, N3, N4, 及びN5は、0又は任意の正の整数でなければならない。それぞれのパディングワードは、任意の値がとられるようにしてもよい。

次に、ClipInfoについて説明する。図46は、ClipInfoのシンタクスを示す図である。ClipInfo()は、それに対応するAVストリームファイル (Clip AVストリーム又はBridge-Clip AVストリームファイル) の属性情報をストアする。

図46に示したClipInfoのシンタクスについて説明するに、version_numberは、このClipInfo()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からClipInfo()の最後までのClipInfo()のバイト数を示す32ビットの符号なし整数である。Clip_stream_typeの8ビットのフィールドは、図47に示すように、Clip Informationファイルに対応するAVストリームのタイプを示す。それぞれのタイプのAVストリームのストリームタイプについては後述する。

offset_SPNの32ビットのフィールドは、AVストリーム (Clip AVストリーム又はBridge-Clip AVストリーム) ファイルの最初のソースパケットについてのソースパケット番号のオフセット値を与える。AVストリームファイルが最初にディスクに記録される時、このoffset_SPNは0でなければならない。

図48に示すように、AVストリームファイルのはじめの部分が編集によって

消去された時、offset_SPNは、0以外の値をとってもよい。本例では、offset_SPNを参照する相対ソースパケット番号（相対アドレス）が、しばしば、RSPN_xxx（xxxは変形する。例、RSPN_EP_start）の形式でシンタクスの中に記述されている。相対ソースパケット番号は、ソースパケット番号を単位とする大きさであり、A Vストリームファイルの最初のソースパケットからoffset_SPNの値を初期値としてカウントされる。

A Vストリームファイルの最初のソースパケットから相対ソースパケット番号で参照されるソースパケットまでのソースパケットの数（SPN_xxx）は、次式で算出される。

$$\text{SPN_xxx} = \text{RSPN_xxx} - \text{offset_SPN}$$

図48に、offset_SPNが4である場合の例を示す。

TS_recording_rateは、24ビットの符号なし整数であり、この値は、DVRドライブ（書込部22）へ又はDVRドライブ（読出部28）からのA Vストリームの必要な入出力のビットレートを与える。record_time_and_dateは、Clipに対応するA Vストリームが記録された時の日時をストアする56ビットのフィールドであり、年／月／日／時／分／秒について、14個の数字を4ビットのBinary Coded Decimal（BCD）で符号化したものである。例えば、2001/12/23:01:02:03は、"0x20011223010203"と符号化される。

durationは、Clipの総再生時間をアライバルタイムクロックに基づいた時間／分／秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary Coded Decimal（BCD）で符号化したものである。例えば、01:45:30は、"0x014530"と符号化される。

time_controlled_flagのフラグは、A Vストリームファイルの記録モードを示す。このtime_controlled_flagが1である場合、記録モードは、記録してからの時間経過に対してファイルサイズが比例するようにして記録されるモードであることを示し、次式に示す条件を満たさなければならない。

$$\text{TS_average_rate} * 192 / 188 * (t - \text{start_time}) - \alpha \leq \text{size_Clip}(t)$$

$$\leq \text{TS_average_rate} * 192 / 188 * (t - \text{start_time}) + \alpha$$

ここで、TS_average_rateは、A Vストリームファイルのトランスポートストリ

ームの平均ビットレートをbytes/second の単位で表したものである。

また、上式において、 t は、秒単位で表される時間を示し、start_timeは、A Vストリームファイルの最初のソースパケットが記録された時の時刻であり、秒単位で表される。size_Clip(t)は、時刻 t におけるA Vストリームファイルのサイズをバイト単位で表したものであり、例えば、start_timeから時刻 t までに10個のソースパケットが記録された場合、size_Clip(t)は10*192バイトである。 α は、TS_average_rateに依存する定数である。

time_controlled_flagが0にセットされている場合、記録モードは、記録の時間経過とA Vストリームのファイルサイズが比例するように制御していないことを示す。例えば、これは入力トランスポートストリームをトランスペアレント記録する場合である。

TS_average_rateは、time_controlled_flagが1にセットされている場合、この24ビットのフィールドは、上式で用いているTS_average_rateの値を示す。time_controlled_flagが0にセットされている場合、このフィールドは、何も意味を持たず、0にセットされなければならない。例えば、可変ビットレートのトランスポートストリームは、次に示す手順により符号化される。まずトランスポートレートをTS_recording_rateの値にセットする。次に、ビデオストリームを可変ビットレートで符号化する。そして、ヌルパケットを使用しないことによって、間欠的にトランスポートパケットを符号化する。

RSPN_arrival_time_discontinuityの32ビットフィールドは、Bridge-Clip A V streamファイル上でアライバルタイムベースの不連続が発生する場所の相対アドレスである。RSPN_arrival_time_discontinuityは、ソースパケット番号を単位とする大きさであり、Bridge-Clip A V streamファイルの最初のソースパケットからClipInfo() において定義されるoffset_SPNの値を初期値としてカウントされる。そのBridge-Clip A V streamファイルの中での絶対アドレスは、上述した

$$\text{SPN}_{xxx} = \text{RSPN}_{xxx} - \text{offset_SPN}$$

に基づいて算出される。

reserved_for_system_useの144ビットのフィールドは、システム用にリザーブされている。is_format_identifier_validのフラグが1である時、format_id

ntifierのフィールドが有効であることを示す。is_original_network_ID_validのフラグが1である場合、original_network_IDのフィールドが有効であることを示す。is_transport_stream_ID_validのフラグが1である場合、transport_stream_IDのフィールドが有効であることを示す。is_servecce_ID_validのフラグが1である場合、servecce_IDのフィールドが有効であることを示す。

is_country_code_validのフラグが1である時、country_codeのフィールドが有効であることを示す。format_identifierの32ビットフィールドは、トランスポートストリームの中でregistration descriptor (ISO/IEC 13818-1で定義されている) が持つformat_identifierの値を示す。original_network_IDの16ビットフィールドは、トランスポートストリームの中で定義されているoriginal_network_IDの値を示す。transport_stream_IDの16ビットフィールドは、トランスポートストリームの中で定義されているtransport_stream_IDの値を示す。

servecce_IDの16ビットフィールドは、トランスポートストリームの中で定義されているservecce_IDの値を示す。country_codeの24ビットのフィールドは、ISO 3166によって定義されるカントリーコードを示す。それぞれのキャラクター文字は、ISO 8859-1で符号化される。例えば、日本は"JPN"と表され、"0x4A 0x50 0x4E"と符号化される。stream_format_nameは、トランスポートストリームのストリーム定義をしているフォーマット機関の名称を示すISO-646の16個のキャラクターコードである。このフィールドの中の無効なバイトは、値'0xFF'がセットされる。

format_identifier、original_network_ID、transport_stream_ID、servecce_ID、country_code、及びstream_format_nameは、トランスポートストリームのサービスプロバイダを示すものであり、これにより、オーディオやビデオストリームの符号化制限、SI(サービスインフォメーション)の規格やオーディオビデオストリーム以外のプライベートデータストリームのストリーム定義を認識することができる。これらの情報は、デコーダが、そのストリームをデコードできるか否か、そしてデコードできる場合にデコード開始前にデコーダシステムの初期設定を行うために用いることが可能である。

次に、STC_Infoについて説明する。ここでは、MPEG-2トランスポートストリームの中でSTCの不連続点（システムタイムベースの不連続点）を含まない時間区間をSTC_sequenceと称し、Clipの中で、STC_sequenceは、STC_sequence_idの値によって特定される。図50A及び図50Bは、連続なSTC区間について説明する図である。同じSTC_sequenceの中で同じSTCの値は、決して現れない（但し、後述するように、Clipの最大時間長は制限されている）。したがって、同じSTC_sequenceの中で同じPTSの値もまた、決して現れない。AVストリームが、 $N(N>0)$ 個のSTC不連続点を含む場合、Clipのシステムタイムベースは、 $(N+1)$ 個のSTC_sequenceに分割される。

STC_Infoは、STCの不連続（システムタイムベースの不連続）が発生する場所のアドレスをストアする。図51を参照して説明するように、RSPN_STC_startが、そのアドレスを示し、最後のSTC_sequenceを除く k 番目（ $k \geq 0$ ）のSTC_sequenceは、 k 番目のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、 $(k+1)$ 番目のRSPN_STC_startで参照されるソースパケットが到着した時刻で終わる。最後のSTC_sequenceは、最後のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、最後のソースパケットが到着した時刻で終了する。

図52は、STC_Infoのシンタクスを示す図である。図52に示したSTC_Infoのシンタクスについて説明するに、version_numberは、このSTC_Info()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からSTC_Info()の最後までのSTC_Info()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI_typeがTU_map typeを示す場合、このlengthフィールドは0をセットしてもよい。CPI()のCPI_typeがEP_map typeを示す場合、num_of_STC_sequencesは1以上の値でなければならない。

num_of_STC_sequencesの8ビットの符号なし整数は、Clipの中でのSTC_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。所定のSTC_sequenceに対応するSTC_sequence_idは、RSPN_STC_startを含むfor-l

oopの中で、そのSTC_sequenceに対応するRSPN_STC_startの現れる順番により定義されるものである。STC_sequence_idは、0から開始される。

RSPN_STC_startの32ビットフィールドは、AVストリームファイル上でSTC_sequenceが開始するアドレスを示す。RSPN_STC_startは、AVストリームファイルの中でシステムタイムベースの不連続点が発生するアドレスを示す。RSPN_STC_startは、AVストリームの中で新しいシステムタイムベースの最初のPCRを持つソースパケットの相対アドレスとしてもよい。RSPN_STC_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、既に上述した

$$\text{SPN_xxx} = \text{RSPN_xxx} - \text{offset_SPN}$$

により算出される。

次に、図45に示したzzzzz.Clipのシンタクス内のProgramInfoについて説明する。図53を参照しながら説明するに、ここでは、Clipの中で次の特徴を持つ時間区間をprogram_sequenceと呼ぶ。先ず、PCR_PIDの値が変わらない。次に、ビデオエレメンタリーストリームの数が増減しない。また、それぞれのビデオストリームについてのPIDの値とそのVideoCodingInfoによって定義される符号化情報が増減しない。更に、オーディオエレメンタリーストリームの数が増減しない。また、それぞれのオーディオストリームについてのPIDの値とそのAudioCodingInfoによって定義される符号化情報が増減しない。

program_sequenceは、同一の時刻において、ただ1つのシステムタイムベースを持つ。program_sequenceは、同一の時刻において、ただ1つのPMTを持つ。ProgramInfo()は、program_sequenceが開始する場所のアドレスをストアする。RSPN_program_sequence_startが、そのアドレスを示す。

図54は、ProgramInfoのシンタクスを示す図である。図54に示したProgramInfoのシンタクスを説明するに、version_numberは、このProgramInfo()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からProgramInfo()の最後までのProgr

amInfo()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI_typeがTU_map typeを示す場合、このlengthフィールドは0にセットされてもよい。CPI()のCPI_typeがEP_map typeを示す場合、number_of_programsは1以上の値でなければならない。

number_of_program_sequencesの8ビットの符号なし整数は、Clipの中でのprogram_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。Clipの中でprogram_sequenceが変化しない場合、number_of_program_sequencesは1をセットされなければならない。RSPN_program_sequence_startの32ビットフィールドは、AVストリームファイル上でプログラムシーケンスが開始する場所の相対アドレスである。

RSPN_program_sequence_startは、ソースバケット番号を単位とする大きさであり、AVストリームファイルの最初のソースバケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

$$\text{SPN_xxx} = \text{RSPN_xxx} - \text{offset_SPN}$$

により算出される。シンタクスのfor-loopの中でRSPN_program_sequence_start値は、昇順に現れなければならない。

PCR_PIDの16ビットフィールドは、そのprogram_sequenceに有効なPCRフィールドを含むトランスポートパケットのPIDを示す。number_of_videosの8ビットフィールドは、video_stream_PIDとVideoCodingInfo()を含むfor-loopのループ回数を示す。number_of_audiosの8ビットフィールドは、audio_stream_PIDとAudioCodingInfo()を含むfor-loopのループ回数を示す。video_stream_PIDの16ビットフィールドは、そのprogram_sequenceに有効なビデオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くVideoCodingInfo()は、そのvideo_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

audio_stream_PIDの16ビットフィールドは、そのprogram_sequenceに有効なオーディオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くAudioCodingInfo()は、そのaudio_stream_PIDで参照されるビデオス

トリームの内容を説明しなければならない。

なお、シンタクスのfor-loopの中でvideo_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でビデオストリームのPIDが符号化されている順番に等しくなければならない。また、シンタクスのfor-loopの中でaudio_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でオーディオストリームのPIDが符号化されている順番に等しくなければならない。

図55は、図54に示したProgramInfoのシンタクス内のVideoCodingInfoのシンタクスを示す図である。図55に示したVideoCodingInfoのシンタクスを説明するに、video_formatの8ビットフィールドは、図56に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオフォーマットを示す。

frame_rateの8ビットフィールドは、図57に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオのフレームレートを示す。display_aspect_ratioの8ビットフィールドは、図58に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオの表示アスペクト比を示す。

図59は、図54に示したProgramInfoのシンタクス内のAudioCodingInfoのシンタクスを示す図である。図59に示したAudioCodingInfoのシンタクスを説明するに、audio_codingの8ビットフィールドは、図60に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオの符号化方法を示す。

audio_component_typeの8ビットフィールドは、図61に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのコンポーネントタイプを示す。sampling_frequencyの8ビットフィールドは、図62に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのサンプリング周波数を示す。

次に、図45に示したzzzzz.Clipのシンタクス内のCPI (Characteristic Point Information)について説明する。CPIは、AVストリームの中の時間情報とそのファイルの中のアドレスとを関連づけるためにある。CPIには2つのタイプがあり、これらはEP_mapとTU_mapである。図63に示すように、CPI()の中のCPI_typeがEP_map typeの場合、そのCPI()はEP_mapを含む。図64に示すように、CPI()の中

のCPI_typeがTU_map typeの場合、そのCPI()はTU_mapを含む。1つのAVストリームは、1つのEP_map又は1つのTU_mapを持つ。AVストリームがSESFトランスポートストリームの場合、それに対応するClipはEP_mapを持たなければならない。

図65は、CPIのシンタクスを示す図である。図65に示したCPIのシンタクスを説明するに、version_numberは、このCPI()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からCPI()の最後までCPI()のバイト数を示す32ビットの符号なし整数である。CPI_typeは、図66に示すように、1ビットのフラグであり、ClipのCPIのタイプを表す。

次に、図65に示したCPIのシンタクス内のEP_mapについて説明する。EP_mapには、2つのタイプがあり、それはビデオストリーム用のEP_mapとオーディオストリーム用のEP_mapである。EP_mapの中のEP_map_typeが、EP_mapのタイプを区別する。Clipが1つ以上のビデオストリームを含む場合、ビデオストリーム用のEP_mapが使用されなければならない。Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、オーディオストリーム用のEP_mapが使用されなければならない。

ビデオストリーム用のEP_mapについて図67を参照して説明する。ビデオストリーム用のEP_mapは、stream_PID、PTS_EP_start、及び、RSPN_EP_startというデータを持つ。stream_PIDは、ビデオストリームを伝送するトランスポートパケットのPIDを示す。PTS_EP_startは、ビデオストリームのシーケンスヘッダから始まるアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットの第1バイト目を含むソースパケットのアドレスを示す。

EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランスポートパケットによって伝送されるビデオストリーム毎に作られる。Clipの中に複数のビデオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID()を含んでもよい。

オーディオストリーム用のEP_mapは、stream_PID、PTS_EP_start、及びRSPN_E

P_startというデータを持つ。stream_PIDは、オーディオストリームを伝送するトランスポートパケットのPIDを示す。PTS_EP_startは、オーディオストリームのアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startで参照されるアクセスユニットの第1バイト目を含むソースパケットのアドレスを示す。

EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランスポートパケットによって伝送されるオーディオストリーム毎に作られる。Clipの中に複数のオーディオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID()を含んでもよい。

EP_mapとSTC_Infoの関係を説明するに、1つのEP_map_for_one_stream_PID()は、STCの不連続点に関係なく1つのテーブルに作られる。RSPN_EP_startの値とSTC_Info()において定義されるRSPN_STC_startの値を比較することにより、それぞれのSTC_sequenceに属するEP_mapのデータの境界が分かる(図68を参照)。EP_mapは、同じPIDで伝送される連続したストリームの範囲に対して、1つのEP_map_for_one_stream_PIDを持たねばならない。図69に示したような場合、program#1とprogram#3は、同じビデオPIDを持つが、データ範囲が連続していないので、それぞれのプログラム毎にEP_map_for_one_stream_PIDを持たねばならない。

図70は、EP_mapのシンタクスを示す図である。図70に示したEP_mapのシンタクスを説明するに、EP_typeは、4ビットのフィールドであり、図71に示すように、EP_mapのエントリポイントタイプを示す。EP_typeは、このフィールドに続くデータフィールドのセマンティクスを示す。Clipが1つ以上のビデオストリームを含む場合、EP_typeは0('video')にセットされなければならない。又は、Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、EP_typeは1('audio')にセットされなければならない。

number_of_stream_PIDsの16ビットのフィールドは、EP_map()の中のnumber_of_stream_PIDsを変数に持つfor-loopのループ回数を示す。stream_PID(k)の16ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるk番目のエレメンタリーストリーム(ビデオ又はオーディオストリ

ーム) を伝送するトランスポートパケットのPIDを示す。EP_typeが0 ('video') に等しい場合、そのエレメンタリストリームはビデオストリームでなければならない。また、EP_typeが1 ('audio') に等しい場合、そのエレメンタリストリームはオーディオストリームでなければならない。

num_EP_entries(k)の16ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるnum_EP_entries(k)を示す。EP_map_for_one_stream_PID_Start_address(k): この32ビットのフィールドは、EP_map()の中でEP_map_for_one_stream_PID(num_EP_entries(k))が始まる相対バイト位置を示す。この値は、EP_map()の第1バイト目からの大きさで示される。

padding_wordは、EP_map()のシンタクスにしたがって挿入されなければならない。XとYは、0又は任意の正の整数でなければならない。それぞれのパディングワードは、任意の値を取ってもよい。

図7.2は、EP_map_for_one_stream_PIDのシンタクスを示す図である。図7.2に示したEP_map_for_one_stream_PIDのシンタクスを説明するに、PTS_EP_startの32ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0 ('video') に等しい場合、このフィールドは、ビデオストリームのシーケンスヘッダで始まるアクセスユニットの33ビット精度のPTSの上位32ビットを持つ。EP_typeが1 ('audio') に等しい場合、このフィールドは、オーディオストリームのアクセスユニットの33ビット精度のPTSの上位32ビットを持つ。

RSPN_EP_startの32ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0 ('video') に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのシーケンスヘッダの第1バイト目を含むソースパケットの相対アドレスを示す。又は、EP_typeが1 ('audio') に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのオーディオフレームの第1バイト目を含むソースパケットの相対アドレスを示す。

RSPN_EP_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset

_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

$$\text{SPN_xxx} = \text{RSPN_xxx} - \text{offset_SPN}$$

により算出される。シンタクスのfor-loopの中でRSPN_EP_startの値は、昇順に現れなければならない。

次に、TU_mapについて、図73を参照して説明する。TU_mapは、ソースパケットのアライバルタイムクロック（到着時刻ベースの時計）に基づいて、1つの時間軸を作る。その時間軸は、TU_map_time_axisと呼ばれる。TU_map_time_axisの原点は、TU_map()の中のoffset_timeによって示される。TU_map_time_axisは、offset_timeから一定の単位に分割される。その単位を、time_unitという。

AVストリームの中の各々のtime_unitの中で、最初の完全な形のソースパケットのAVストリームファイル上のアドレスが、TU_mapにストアされる。これらのアドレスを、RSPN_time_unit_startと称する。TU_map_time_axis上において、k (k>=0)番目のtime_unitが始まる時刻は、TU_start_time(k)と呼ばれる。この値は次式に基づいて算出される。

$$\text{TU_start_time}(k) = \text{offset_time} + k * \text{time_unit_size}$$

TU_start_time(k)は、45kHzの精度を持つ。

図74は、TU_mapのシンタクスを示す図である。図74に示したTU_mapのシンタクスを説明するに、offset_timeの32ビット長のフィールドは、TU_map_time_axisに対するオフセットタイムを与える。この値は、Clipの中の最初のtime_unitに対するオフセット時刻を示す。offset_timeは、27MHz精度のアライバルタイムクロックから導き出される45kHzクロックを単位とする大きさである。AVストリームが新しいClipとして記録される場合、offset_timeは0にセットされなければならない。

time_unit_sizeの32ビットフィールドは、time_unitの大きさを与えるものであり、それは27MHz精度のアライバルタイムクロックから導き出される45kHzクロックを単位とする大きさである。time_unit_sizeは、1秒以下 (time_unit_size<=45000) にすることがよい。number_of_time_unit_entriesの32ビットフィールドは、TU_map()の中にストアされているtime_unitのエントリ数を示

す。

RSPN_time_unit_startの32ビットフィールドは、AVストリームの中でそれぞれのtime_unitが開始する場所の相対アドレスを示す。RSPN_time_unit_startは、ソースパケット番号を単位とする大きさであり、AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、

$$\text{SPN_xxx} = \text{RSPN_xxx} - \text{offset_SPN}$$

により算出される。シンタクスのfor-loopの中でRSPN_time_unit_startの値は、昇順に現れなければならない。(k+1)番目のtime_unitの中にソースパケットが何もない場合、(k+1)番目のRSPN_time_unit_startは、k番目のRSPN_time_unit_startと等しくなければならない。

図45に示したzzzzz.Clipのシンタクス内のClipMarkについて説明する。ClipMarkは、クリップについてのマーク情報であり、ClipMarkの中にストアされる。このマークは、記録器（記録再生装置1）によってセットされるものであり、ユーザによってセットされるものではない。

図75は、ClipMarkのシンタクスを示す図である。図75に示したClipMarkのシンタクスを説明するに、version_numberは、このClipMark()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からClipMark()の最後までのClipMark()のバイト数を示す32ビットの符号なし整数である。number_of_clip_marksは、ClipMarkの中にストアされているマークの個数を示す16ビットの符号なし整数。number_of_clip_marksは、0であってもよい。mark_typeは、マークのタイプを示す8ビットのフィールドであり、図76に示すテーブルに従って符号化される。

mark_time_stampは、32ビットフィールドであり、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図77に示すように、Playlist()の中のCPI_typeにより異なる。

STC_sequence_idは、CPI()の中のCPI_typeがEP_map typeを示す場合、この8ビットのフィールドは、mark_time_stampが置かれているところのSTC連続区間の

STC_sequence_idを示す。CPI()の中のCPI_typeがTU_map typeを示す場合、この8ビットのフィールドは何も意味を持たず、0にセットされる。character_setの8ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示される値に対応する。

name_lengthの8ビットフィールドは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。mark_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていてもよい。

ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのマークにはサムネイル画像が付加されていない。

図78は、図75に代わるClipMarkの他のシンタクスを示す図であり、図79は、その場合における、図76に代わるmark_typeのテーブルの例を示す。reserved_for_maker_IDは、mark_typeが、0xC0から0xFFの値を示す時に、そのmark_typeを定義しているメーカーのメーカーIDを示す16ビットのフィールドである。メーカーIDは、DVRフォーマットライセンサが指定する。mark_entry()は、マーク点に指定されたポイントを示す情報であり、そのシンタクスの詳細は後述する。representative_picture_entry()は、mark_entry()によって示されるマークを代表する画像のポイントを示す情報であり、そのシンタクスの詳細は後述する。

ClipMarkは、ユーザがAVストリームを再生するときに、その内容を視覚的に検索できるようにするために用いられる。DVRプレーヤは、GUI(グラフィカルユーザインタフェース)を使用して、ClipMarkの情報をユーザに提示する。ClipMarkの情報を視覚的に表示するためには、mark_entry()が示すピクチャよりもむしろrepresentative_picture_entry()が示すピクチャを示したほうがよい。

図80に、mark_entry()とrepresentative_picture_entry()の例を示す。例え

ば、あるプログラムが開始してから、しばらくした後（数秒後）、そのプログラムの番組名（タイトル）が表示されれるとする。ClipMarkを作るときは、mark_entry()は、そのプログラムの開始ポイントに置き、representative_picture_entry()は、そのプログラムの番組名（タイトル）が表示されるポイントに置くようにしてもよい。

D V R プレーヤは、representative_picture_entryの画像を G U I に表示し、ユーザがその画像を指定すると、D V R プレーヤは、mark_entryの置かれたポイントから再生を開始する。

mark_entry() 及び representative_picture_entry()のシンタクスを、図 8 1 に示す。

mark_time_stampは、32ビットフィールドであり、mark_entry()の場合はマークが指定されたポイントを示すタイムスタンプをストアし、またrepresentative_picture_entry()の場合、mark_entry()によって示されるマークを代表する画像のポイントを示すタイムスタンプをストアする。

次に、ClipMarkを指定するために、P T Sによるタイムスタンプベースの情報を使用するのではなく、アドレスベースの情報を使用する場合のmark_entry()とrepresentative_picture_entry()のシンタクスの例を図 8 2 に示す。

RSPN_ref_EP_startは、mark_entry()の場合、A Vストリームの中でマーク点のピクチャをデコードするためのストリームのエントリポイントを示すソースパケットの相対アドレスを示す。また、representative_picture_entry()の場合、mark_entry()によって示されるマークを代表するピクチャをデコードするためのストリームのエントリポイントを示すソースパケットの相対アドレスを示す。RSPN_ref_EP_startの値は、EP_mapの中にRSPN_EP_startとしてストアされていなければならない、且つ、そのRSPN_EP_startに対応するPTS_EP_startの値は、EP_mapの中で、マーク点のピクチャのP T Sより過去で最も近い値でなければならない。

offset_num_picturesは、32ビットのフィールドであり、RSPN_ref_EP_startにより参照されるピクチャから表示順序でマーク点で示されるピクチャまでのオフセットのピクチャ数を示す。この数は、0からカウントされる。図 8 3 の例の場合、offset_num_picturesは6となる。

次に、ClipMarkを指定するために、アドレスベースの情報をを使用する場合のmark_entry() と representative_picture_entry()のシンタクスの別の例を図84に示す。

RSPN_mark_pointは、mark_entry()の場合、AVストリームの中で、そのマークが参照するアクセスユニットの第1バイト目を含むソースパケットの相対アドレスを示す。また、representative_picture_entry()の場合、mark_entry()によって示されるマークを代表する符号化ピクチャの第1バイト目を含むソースパケットの相対アドレスを示す。

RSPN_mark_pointは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClip Information fileにおいて定義されるoffset_SPNの値を初期値としてカウントされる。

図85を用いて、ClipMarkとEP_mapの関係を説明する。この例の場合、EP_mapが、エントリポイントのアドレスとしてI0, I1, Inを指定しており、これらのアドレスからシーケンスヘッダに続くIピクチャが開始しているとする。ClipMarkが、あるマークのアドレスとして、M1を指定している時、そのソースパケットから開始しているピクチャをデコードできるためには、M1のアドレスより前で最も近いエントリポイントであるI1からデータを読み出し開始すればよい。

MakersPrivateDataについては、図22を参照して既に説明したので、その説明は省略する。

次に、サムネイルインフォメーション (Thumbnail Information) について説明する。サムネイル画像は、menu.thmbファイル又はmark.thmbファイルにストアされる。これらのファイルは同じシンタクス構造であり、ただ1つのThumbnail()を持つ。menu.thmbファイルは、メニューサムネイル画像、すなわちVolumeを代表する画像、及び、それぞれのPlayListを代表する画像をストアする。全てのメニューサムネイルは、ただ1つのmenu.thmbファイルにストアされる。

mark.thmbファイルは、マークサムネイル画像、すなわちマーク点を表すピクチャをストアする。全てのPlayList及びClipに対する全てのマークサムネイルは、ただ1つのmark.thmbファイルにストアされる。サムネイルは頻繁に追加、削除さ

れるので、追加操作と部分削除の操作は容易に高速に実行できなければならない。この理由のため、Thumbnail()はブロック構造を有する。画像のデータはいくつかの部分に分割され、各部分は1つのtn_blockに格納される。1つの画像データは連続したtn_blockに格納される。tn_blockの列には、使用されていないtn_blockが存在してもよい。1つのサムネイル画像のバイト長は可変である。

図86は、menu.thmbとmark.thmbのシンタクスを示す図であり、図87は、図86に示したmenu.thmbとmark.thmbのシンタクス内のThumbnailのシンタクスを示す図である。図87に示したThumbnailのシンタクスについて説明するに、version_numberは、このThumbnail()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からThumbnail()の最後までのMakersPrivateData()のバイト数を示す32ビットの符号なし整数である。tn_blocks_start_addressは、Thumbnail()の先頭のバイトからの相対バイト数を単位として、最初のtn_blockの先頭バイトアドレスを示す32ビットの符号なし整数である。相対バイト数は0からカウントされる。number_of_thumbnailsは、Thumbnail()の中に含まれているサムネイル画像のエントリ数を与える16ビットの符号なし整数である。

tn_block_sizeは、1024バイトを単位として、1つのtn_blockの大きさを与える16ビットの符号なし整数である。例えば、tn_block_size=1ならば、それは1つのtn_blockの大きさが1024バイトであることを示す。number_of_tn_blocksは、このThumbnail()中のtn_blockのエントリ数を表す116ビットの符号なし整数である。thumbnail_indexは、このthumbnail_indexフィールドから始まるforループ一回分のサムネイル情報で表されるサムネイル画像のインデクス番号を表す16ビットの符号なし整数である。thumbnail_indexとして、0xFFFFという値を使用してはならない。thumbnail_indexはUIAppInfoVolume()、UIAppInfoPlaylist()、PlaylistMark()、及びClipMark()の中のref_thumbnail_indexによって参照される。

thumbnail_picture_formatは、サムネイル画像のピクチャフォーマットを表す

8ビットの符号なし整数で、図88に示すような値をとる。表中のDCFとPNGは”menu.thmb”内でのみ許される。マークサムネイルは、値”0x00”(MPEG-2 Video I-picture)をとらなければならない。

picture_data_sizeは、サムネイル画像のバイト長をバイト単位で示す32ビットの符号なし整数である。start_tn_block_numberは、サムネイル画像のデータが始まるtn_blockのtn_block番号を表す16ビットの符号なし整数である。サムネイル画像データの先頭は、tb_blockの先頭と一致していなければならない。tn_block番号は、0から始まり、tn_blockのfor-ループ中の変数kの値に関係する。

x_picture_lengthは、サムネイル画像のフレーム画枠の水平方向のピクセル数を表す16ビットの符号なし整数である。y_picture_lengthは、サムネイル画像のフレーム画枠の垂直方向のピクセル数を表す16ビットの符号なし整数である。tn_blockは、サムネイル画像がストアされる領域である。Thumbnail()の中の全てのtn_blockは、同じサイズ(固定長)であり、その大きさはtn_block_sizeによって定義される。

図89A及び図89Bは、サムネイル画像データがどのようにtn_blockに格納されるかを模式的に表した図である。図89A及び図89Bのように、各サムネイル画像データはtn_blockの先頭から始まり、1 tn_blockを超える大きさの場合は、連続する次のtn_blockを使用してストアされる。このようにすることにより、可変長であるピクチャデータが、固定長のデータとして管理することが可能となり、削除といった編集に対して簡便な処理により対応することができるようになる。

次に、AVストリームファイルについて説明する。AVストリームファイルは、”M2TS”ディレクトリ(図14)にストアされる。AVストリームファイルには、2つのタイプがあり、これらは、Clip AVストリームとBridge-Clip AVストリームファイルである。両方のAVストリーム共に、これ以降で定義されるDVR MPEG-2トランスポートストリームファイルの構造でなければならない。

まず、DVR MPEG-2トランスポートストリームについて説明する。DVR MPEG-2トランスポートストリームの構造は、図90に示すようになっている。AVストリームファイルは、DVR MPEG-2トランスポートスト

リームの構造を持つ。DVR MPEG-2 トラnsポートストリームは、整数個のAligned unitから構成される。Aligned unitの大きさは、6144 バイト (2048*3 バイト)である。Aligned unitは、ソースパケットの第1バイト目から始まる。ソースパケットは、192バイト長である。1つのソースパケットは、TP_extra_headerとトラnsポートパケットからなる。TP_extra_headerは、4バイト長であり、またトラnsポートパケットは、188バイト長である。

1つのAligned unitは、32個のソースパケットからなる。DVR MPEG-2 トラnsポートストリームの中の最後のAligned unitも、また32個のソースパケットからなる。よって、DVR MPEG-2 トラnsポートストリームは、Aligned unitの境界で終端する。ディスクに記録される入力トラnsポートストリームのトラnsポートパケットの数が32の倍数でない時、ヌルパケット (PID=0x1FFFのトラnsポートパケット)を持ったソースパケットを最後のAligned unitに使用しなければならない。ファイルシステムは、DVR MPEG-2 トラnsポートストリームに余分な情報を付加してはならない。

図91に、DVR MPEG-2 トラnsポートストリームのレコーダモデルを示す。図91に示したレコーダは、レコーディングプロセスを規定するための概念上のモデルである。DVR MPEG-2 トラnsポートストリームは、このモデルに従う。

MPEG-2 トラnsポートストリームの入力タイミングについて説明する。入力MPEG-2 トラnsポートストリームは、フルトラnsポートストリーム又はパーシャルトラnsポートストリームである。入力されるMPEG-2 トラnsポートストリームは、ISO/IEC 13818-1又はISO/IEC 13818-9に従っていなければならない。MPEG-2 トラnsポートストリームのi番目のバイトは、T-STD (ISO/IEC 13818-1で規定されるTransport stream system target decoder) 5.1とソースパケットタイザ (source packetizer) 5.4へ、時刻t (i) に同時に入力される。Rpkは、トラnsポートパケットの入力レート of 瞬時的な最大値である。

27 MHz PLL 5.2は、27 MHzクロックの周波数を発生する。27 MHzクロックの周波数は、MPEG-2 トラnsポートストリームのPCR (Prog

ram Clock Reference)の値にロックされる。アライバルタイムクロックカウンタ (arrival time clock counter) 53は、27MHzの周波数のパルスをカウントするバイナリーカウンタである。Arrival_time_clock(i)は、時刻t(i)におけるarrival time clock counter 53のカウント値である。

source packetizer 54は、全てのトランスポートパケットにTP_extra_headerを付加し、ソースパケットを作る。Arrival_time_stampは、トランスポートパケットの第1バイト目がTS 51とソースパケットタイザ 54の両方へ到着する時刻を表す。Arrival_time_stamp(k)は、次式で示されるようにArrival_time_clock(k)のサンプル値であり、ここで、kはトランスポートパケットの第1バイト目を示す。

$$\text{arrival_time_stamp}(k) = \text{arrival_time_clock}(k) \% 230$$

2つの連続して入力されるトランスポートパケットの時間間隔が、230/27000000秒(約40秒)以上になる場合、その2つのトランスポートパケットのarrival_time_stampの差分は、230/27000000秒になるようにセットされるべきである。レコーダは、そのようになる場合に備えてある。

スムージングバッファ (smoothing buffer) 55は、入力トランスポートストリームのビットレートをスムージングする。スムージングバッファ 55は、オーバーフローしてはならない。Rmaxは、スムージングバッファ 55が空でない時のスムージングバッファ 55からのソースパケットの出力ビットレートである。スムージングバッファ 55が空である時、スムージングバッファ 55からの出力ビットレートは0である。

次に、DVR MPEG-2 トランスポートストリームのレコーダモデルのパラメータについて説明する。Rmaxという値は、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateによって与えられる。この値は、次式により算出される。

$$R_{\max} = \text{TS_recording_rate} * 192/188$$

TS_recording_rateの値は、bytes/secondを単位とする大きさである。

入力トランスポートストリームがSESFトランスポートストリームの場合、Rpkは、AVストリームファイルに対応するClipInfo()において定義されるTS_recording

_rateに等しくなければならない。入力トランスポートストリームがSESFトランスポートストリームでない場合、この値はMPEG-2 transport streamのデスクリプター、例えばmaximum_bitrate_descriptorやpartial_transport_stream_descriptorなど、において定義される値を参照してもよい。

入力トランスポートストリームがSESFトランスポートストリームの場合、スムージングバッファ55の大きさ(smoothing buffer size)は0である。入力トランスポートストリームがSESFトランスポートストリームでない場合、スムージングバッファ55の大きさはMPEG-2 transport streamのデスクリプター、例えばsmoothing_buffer_descriptor、short_smoothing_buffer_descriptor、partial_transport_stream_descriptorなどにおいて定義される値を参照してもよい。

記録機(レコーダ)及び記録再生装置1(プレーヤ)は、十分なサイズのバッファを用意しなければならない。デフォルトのバッファサイズは、1536バイトである。

次に、DVR MPEG-2 トランスポートストリームのプレーヤモデルについて説明する。図92は、DVR MPEG-2 トランスポートストリームのプレーヤモデルを示す図である。これは、再生プロセスを規定するための概念上のモデルである。DVR MPEG-2 トランスポートストリームは、このモデルに従う。

27MHz Xtal(クリスタル発振器)61は、27MHzの周波数を発生する。27MHz周波数の誤差範囲は、 ± 30 ppm (27000000 \pm 810 Hz)でなければならない。arrival time clock counter 62は、27MHzの周波数のパルスのカウントするバイナリーカウンタである。arrival_time_clock(i)は、時刻t(i)におけるarrival time clock counter 62のカウント値である。

smoothing buffer 64において、Rmaxは、スムージングバッファ64がフルでない時のスムージングバッファ64へのソースパケットの入力ビットレートである。スムージングバッファ64がフルである時、スムージングバッファ64への入力ビットレートは0である。

MPEG-2 トランスポートストリームの出力タイミングを説明するに、現在のソースパケットのarrival_time_stampがarrival_time_clock(i)のLSB 30ビットの値と等しい時、そのソースパケットのトランスポートパケットは、スム

ージングバッファ 64 から引き抜かれる。Rpkは、トランスポートパケットレートの瞬時的な最大値である。スムージングバッファ 64 は、アンダーフローしてはならない。

D V R M P E G - 2 トランスポートストリームのプレーヤモデルのパラメータについては、上述した D V R M P E G - 2 トランスポートストリームのレコーダモデルのパラメータと同一である。

図 9 3 は、Source packetのシンタクスを示す図である。transport_packet() は、I S O / I E C 1 3 8 1 8 - 1 で規定される M P E G - 2 トランスポートパケットである。図 9 3 に示した Source packetのシンタクス内の TP_Extra_header のシンタクスを図 9 4 に示す。図 9 4 に示した TP_Extra_header のシンタクスについて説明するに、copy_permission_indicator は、トランスポートパケットのペイロードのコピー制限を表す整数である。コピー制限は、copy free、no more copy、copy once、又は copy prohibited とすることができる。図 9 5 は、copy_permission_indicator の値と、それらによって指定されるモードの関係を示す。

copy_permission_indicator は、全てのトランスポートパケットに付加される。I E E E 1 3 9 4 デジタルインタフェースを使用して入力トランスポートストリームを記録する場合、copy_permission_indicator の値は、IEEE1394 isochronous packet header 中の EMI (Encryption Mode Indicator) の値に関連付けてもよい。I E E E 1 3 9 4 デジタルインタフェースを使用しないで入力トランスポートストリームを記録する場合、copy_permission_indicator の値は、トランスポートパケットの中に埋め込まれた C C I の値に関連付けてもよい。アナログ信号入力をセルフエンコードする場合、copy_permission_indicator の値は、アナログ信号の C G M S - A の値に関連付けてもよい。

arrival_time_stamp は、次式

$$\text{arrival_time_stamp}(k) = \text{arrival_time_clock}(k) \% 230$$

において、arrival_time_stamp によって指定される値を持つ整数値である。

Clip A V ストリームの定義をするに、Clip A V ストリームは、上述したような定義がされる D V R M P E G - 2 トランスポートストリームの構造を持たねばならない。arrival_time_clock(i) は、Clip A V ストリームの中で連続して増加

しなければならない。Clip AVストリームの中にシステムタイムベース（STCベース）の不連続点が存在したとしても、そのClip AVストリームのarrival_time_clock(i)は、連続して増加しなければならない。

Clip AVストリームの中の開始と終了の間のarrival_time_clock(i)の差分の最大値は、26時間でなければならない。この制限は、MP EG-2トランスポートストリームの中にシステムタイムベース（STCベース）の不連続点が存在しない場合に、Clip AVストリームの中で同じ値のPTS(Presentation Time Stamp)が決して現れないことを保証する。MP EG-2システムズ規格は、PTSのラップアラウンド周期を233/90000秒(約26.5時間).と規定している。

Bridge-Clip AVストリームの定義をするに、Bridge-Clip AVストリームは、上述したような定義がされるDVR MP EG-2トランスポートストリームの構造を持たねばならない。Bridge-Clip AVストリームは、1つのアライバルタイムベースの不連続点を含まなければならない。アライバルタイムベースの不連続点の前後のトランスポートストリームは、後述する符号化の制限に従わなければならない。且つ後述するDVR-STDに従わなければならない。

本例においては、編集におけるPlayItem間のビデオとオーディオのシームレス接続をサポートする。PlayItem間をシームレス接続にすることは、プレーヤ/レコーダに”データの連続供給”と”シームレスな復号処理”を保証する。”データの連続供給”とは、ファイルシステムが、デコーダにバッファのアンダーフローを起こさせることのないように必要なビットレートでデータを供給することを保証できることである。データのリアルタイム性を保証して、データをディスクから読み出すことができるように、データが十分な大きさの連続したブロック単位でストアされるようにする。

”シームレスな復号処理”とは、プレーヤが、デコーダの再生出力にポーズやギャップを起こさせることなく、ディスクに記録されたオーディオビデオデータを表示できることである。

シームレス接続されているPlayItemが参照するAVストリームについて説明する。先行するPlayItemと現在のPlayItemの接続が、シームレス表示できるように保証されているかどうかは、現在のPlayItemにおいて定義されているconnection

_conditionフィールドから判断することができる。PlayItem間のシームレス接続は、Bridge-Clipを使用する方法と使用しない方法がある。

図9 6は、Bridge-Clipを使用する場合の先行するPlayItemと現在のPlayItemの関係を示している。図9 6においては、プレーヤが読み出すストリームデータが、影を付けて示されている。図9 6に示したTS 1は、Clip1 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータからなる。

TS 1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time (図9 6においてIN_time1で図示されている) に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスから、RSPN_exit_from_previous_Clipで参照されるソースパケットまでのストリームデータである。TS 1に含まれるBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータは、Bridge-Clipの最初のソースパケットから、RSPN_arrival_time_discontinuityで参照されるソースパケットの直前のソースパケットまでのストリームデータである。

また、図9 6におけるTS 2は、Clip2 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータからなる。TS 2に含まれるBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータは、RSPN_arrival_time_discontinuityで参照されるソースパケットから、Bridge-Clipの最後のソースパケットまでのストリームデータである。TS 2のClip2の影を付けられたストリームデータは、RSPN_enter_to_current_Clipで参照されるソースパケットから、現在のPlayItemのOUT_time (図9 6においてOUT_time2で図示されている) に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスまでのストリームデータである。

図9 7は、Bridge-Clipを使用しない場合の先行するPlayItemと現在のPlayItemの関係を示している。この場合、プレーヤが読み出すストリームデータは、影を付けて示されている。図9 7におけるTS 1は、Clip1 (Clip AVストリーム) の影を付けられたストリームデータからなる。TS 1のClip1の影を付けられたスト

リームデータは、先行するPlayItemのIN_time（図97においてIN_time1で図示されている）に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスから始まり、Clip1の最後のソースバケットまでのデータである。また、図97におけるTS2は、Clip2（Clip AVストリーム）の影を付けられたストリームデータからなる。

TS2のClip2の影を付けられたストリームデータは、Clip2の最初のソースバケットから始まり、現在のPlayItemのOUT_time（図97においてOUT_time2で図示されている）に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスまでのストリームデータである。

図96と図97において、TS1とTS2は、ソースバケットの連続したストリームである。次に、TS1とTS2のストリーム規定と、それらの間の接続条件について考える。まず、シームレス接続のための符号化制限について考える。トランスポートストリームの符号化構造の制限として、まず、TS1とTS2の中に含まれるプログラムの数は、1でなければならない。TS1とTS2の中に含まれるビデオストリームの数は、1でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、2以下でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、等しくなければならない。TS1及び／又はTS2の中に、上記以外のエレメンタリーストリーム又はプライベートストリームが含まれていてもよい。

ビデオビットストリームの制限について説明する。図98は、ピクチャの表示順序で示すシームレス接続の例を示す図である。接続点においてビデオストリームをシームレスに表示できるためには、OUT_time1（Clip1のOUT_time）の後とIN_time2（Clip2のIN_time）の前に表示される不必要なピクチャは、接続点付近のClipの部分的なストリームを再エンコードするプロセスにより、除去されなければならない。

図98に示したような場合において、BridgeSequenceを使用してシームレス接続を実現する例を、図99に示す。RSPN_arrival_time_discontinuityより前のBridge-Clipのビデオストリームは、図98のClip1のOUT_time1に対応するピクチャまでの符号化ビデオストリームからなる。そして、そのビデオストリームは先

行するClip1のビデオストリームに接続され、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。

同様に、RSPN_arrival_time_discontinuity以後のBridge-Clipのビデオストリームは、図98のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームからなる。そして、そのビデオストリームは、正しくデコード開始することができて、これに続くClip2のビデオストリームに接続され、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。Bridge-Clipを作るためには、一般に、数枚のピクチャは再エンコードしなければならない、それ以外のピクチャはオリジナルのClipからコピーすることができる。

図98に示した例の場合にBridgeSequenceを使用しないでシームレス接続を実現する例を図100に示す。Clip1のビデオストリームは、図98のOUT_time1に対応するピクチャまでの符号化ビデオストリームからなり、それは、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。同様に、Clip2のビデオストリームは、図98のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームからなり、それは、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。

ビデオストリームの符号化制限について説明するに、先ず、TS1とTS2のビデオストリームのフレームレートは、等しくなければならない。TS1のビデオストリームは、sequence_end_codeで終端しなければならない。TS2のビデオストリームは、Sequence Header、GOP Header、そしてIピクチャで開始しなければならない。TS2のビデオストリームは、クローズドGOPで開始しなければならない。

ビットストリームの中で定義されるビデオプレゼンテーションユニット（フレーム又はフィールド）は、接続点を挟んで連続でなければならない。接続点において、フレーム又はフィールドのギャップがあってはならない。接続点において、トップーボトムのフィールドシーケンスは連続でなければならない。3-2プルダウンを使用するエンコードの場合は、"top_field_first" 及び "repeat_first

“_field”フラグを書き換える必要があるかもしれない、又はフィールドギャップの発生を防ぐために局所的に再エンコードするようにしてもよい。

オーディオビットストリームの符号化制限について説明するに、TS 1とTS 2のオーディオのサンプリング周波数は、同じでなければならない。TS 1とTS 2のオーディオの符号化方法（例、MPEG 1レイヤ2、AC-3、SESF LPCM、AAC）は、同じでなければならない。

次に、MPEG-2トランスポートストリームの符号化制限について説明するに、TS 1のオーディオストリームの最後のオーディオフレームは、TS 1の最後の表示ピクチャの表示終了時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。TS 2のオーディオストリームの最初のオーディオフレームは、TS 2の最初の表示ピクチャの表示開始時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。

接続点において、オーディオプレゼンテーションユニットのシーケンスにギャップがあってはならない。図101に示すように、2オーディオフレーム区間未満のオーディオプレゼンテーションユニットの長さで定義されるオーバーラップがあってもよい。TS 2のエレメンタリーストリームを伝送する最初のパケットは、ビデオパケットでなければならない。接続点におけるトランスポートストリームは、後述するDVR-STDに従わなくてはならない。

Clip及びBridge-Clipの制限について説明するに、TS 1とTS 2は、それぞれの中にアライバルタイムベースの不連続点を含んではならない。

以下の制限は、Bridge-Clipを使用する場合にのみ適用される。TS 1の最後のソースパケットとTS 2の最初のソースパケットの接続点においてのみ、Bridge-Clip AVストリームは、ただ1つのアライバルタイムベースの不連続点を持つ。ClipInfo()において定義されるRSPN_arrival_time_discontinuityが、その不連続点のアドレスを示し、それはTS 2の最初のソースパケットを参照するアドレスを示さなければならない。

BridgeSequenceInfo()において定義されるRSPN_exit_from_previous_Clipによって参照されるソースパケットは、Clip 1の中のどのソースパケットでもよい。これは、Aligned unitの境界である必要はない。BridgeSequenceInfo()にお

いて定義されるRSPN_enter_to_current_Clipによって参照されるソースパケットは、Clip 2の中のどのソースパケットでもよい。それは、Aligned unitの境界である必要はない。

PlayItemの制限について説明するに、先行するPlayItemのOUT_time (図9 6、図9 7において示されるOUT_time1) は、TS 1の最後のビデオプレゼンテーションユニットの表示終了時刻を示さなければならない。現在のPlayItemのIN_time (図9 6、図9 7において示されるIN_time2) は、TS 2の最初のビデオプレゼンテーションユニットの表示開始時刻を示さなければならない。

Bridge-Clipを使用する場合のデータアロケーションの制限について、図10 2を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1 (Clip AVストリームファイル) とClip2 (Clip AVストリームファイル) に接続されるBridge-Clip AVストリームを、データアロケーション規定を満たすように配置することによって行われなければならない。

RSPN_exit_from_previous_Clip以前のClip 1 (Clip AVストリームファイル) のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_exit_from_previous_Clipが選択されなければならない。Bridge-Clip AVストリームのデータ長は、ハーフフラグメント以上の連続領域に配置されるように、選択されなければならない。RSPN_enter_to_current_Clip以後のClip 2 (Clip AVストリームファイル) のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_enter_to_current_Clipが選択されなければならない。

Bridge-Clipを使用しないでシームレス接続する場合のデータアロケーションの制限について、図10 3を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1 (Clip AVストリームファイル) の最後の部分とClip2 (Clip AVストリームファイル) の最初の部分を、データアロケーション規定を満たすように配置することによって行われなければならない。

Clip1 (Clip AVストリームファイル) の最後のストリーム部分が、ハーフフ

ラグメント以上の連続領域に配置されていなければならない。Clip2 (Clip A V ストリームファイル) の最初のストリーム部分が、ハーフラグメント以上の連続領域に配置されていなければならない。

次に、DVR-STDについて説明する。DVR-STDは、DVR MPEG-2 トランスポートストリームの生成及び検証の際におけるデコード処理をモデル化するための概念モデルである。また、DVR-STDは、上述したシームレス接続された2つのPlayItemによって参照されるA Vストリームの生成及び検証の際におけるデコード処理をモデル化するための概念モデルでもある。

DVR-STDモデルを図104に示す。図104に示したモデルには、DVR MPEG-2 トランスポートストリームプレーヤモデルが構成要素として含まれている。n, TBn, MBn, EBn, TBSys, Bsys, Rxn, Rbxn, Rxsys, Dn, Dsys, On及びPn(k)の表記方法は、ISO/IEC 13818-1のT-STDに定義されているものと同じである。すなわち、次の通りである。nは、エレメンタリーストリームのインデクス番号である。TBnは、エレメンタリーストリームnのトランスポートバッファである。

MBnは、エレメンタリーストリームnの多重バッファである。ビデオストリームについてのみ存在する。EBnは、エレメンタリーストリームnのエレメンタリーストリームバッファである。ビデオストリームについてのみ存在する。TBSysは、復号中のプログラムのシステム情報のための入力バッファである。Bsysは、復号中のプログラムのシステム情報のためのシステムターゲットデコーダ内のメインバッファである。Rxnは、データがTBnから取り除かれる伝送レートである。Rbxnは、PES パケットペイロードがMBnから取り除かれる伝送レートである。ビデオストリームについてのみ存在する。

Rxsysは、データがTBSysから取り除かれる伝送レートである。Dnは、エレメンタリーストリームnのデコーダである。Dsysは、復号中のプログラムのシステム情報に関するデコーダである。Onは、ビデオストリームnのre-ordering bufferである。Pn(k)は、エレメンタリーストリームnのk番目のプレゼンテーションユニットである。

DVR-STDのデコーディングプロセスについて説明する。単一のDVR M

P E G - 2 トランスポートストリームを再生している間は、トランスポートパケットをTB1, TBn又はTBsysのバッファへ入力するタイミングは、ソースパケットのarrival_time_stampにより決定される。TB1, MB1, EB1, TBn, Bn, TBsys及びBsysのバッファリング動作の規定は、ISO/IEC 13818-1に規定されているT-S-TDと同じである。復号動作と表示動作の規定もまた、ISO/IEC 13818-1に規定されているT-S-TDと同じである。

シームレス接続されたPlayItemを再生している間のデコーディングプロセスについて説明する。ここでは、シームレス接続されたPlayItemによって参照される2つのAVストリームの再生について説明をすることにし、以後の説明では、上述した（例えば、図9.6に示した）TS1とTS2の再生について説明する。TS1は、先行するストリームであり、TS2は、現在のストリームである。

図105は、あるAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移る時のトランスポートパケットの入力、復号、表示のタイミングチャートを示す。所定のAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移る間には、TS2のアライバルタイムベースの時間軸（図105においてATC2で示される）は、TS1のアライバルタイムベースの時間軸（図105においてATC1で示される）と同じでない。

また、TS2のシステムタイムベースの時間軸（図105においてSTC2で示される）は、TS1のシステムタイムベースの時間軸（図105においてSTC1で示される）と同じでない。ビデオの表示は、シームレスに連続していることが要求される。オーディオのプレゼンテーションユニットの表示時間にはオーバーラップがあってもよい。

DVR-S-TDへの入力タイミングについて説明する。時刻T1までの時間、すなわち、TS1の最後のビデオパケットがDVR-S-TDのTB1に入力終了するまでは、DVR-S-TDのTB1、TBn又はTBsysのバッファへの入力タイミングは、TS1のソースパケットのarrival_time_stampによって決定される。

TS1の残りのパケットは、TS_recording_rate (TS1) のビットレートでD

VR-STDのTB_n又はTB_{sys}のバッファへ入力されなければならない。
 ここで、TS_recording_rate(TS1)は、Clip1に対応するClipInfo()において
 定義されるTS_recording_rateの値である。TS1の最後のバイトがバッファへ入
 力する時刻は、時刻T2である。したがって、時刻T1からT2までの区間では、
 ソースパケットのarrival_time_stampは無視される。

N1をTS1の最後のビデオパケットに続くTS1のトランスポートパケット
 のバイト数とすると、時刻T1乃至T2までの時間DT1は、N1バイトがTS_reco
 rding_rate(TS1)のビットレートで入力終了するために必要な時間であり、次
 式により算出される。

$$\Delta T1 = T2 - T1 = N1 / TS_recording_rate(TS1)$$

時刻T1乃至T2までの間は、RX_nとRX_{sys}の値は共に、TS_recording_rate(TS
 1)の値に変化する。このルール以外のバッファリング動作は、T-STDと同じ
 である。

T2の時刻において、arrival time clock counterは、TS2の最初のソースパ
 ケットのarrival_time_stampの値にリセットされる。DVR-STDのTB1、
 TB_n 又はTB_{sys}のバッファへの入力タイミングは、TS2のソースパケット
 のarrival_time_stampによって決定される。RX_nとRX_{sys}は共に、T-STDにお
 いて定義されている値に変化する。

付加的なオーディオバッファリング及びシステムデータバッファリングについ
 て説明するに、オーディオデコーダとシステムデコーダは、時刻T1からT2まで
 の区間の入力データを処理することができるよう、T-STDで定義されるバ
 ッファ量に加えて付加的なバッファ量（約1秒分のデータ量）が必要である。

ビデオのプレゼンテーションタイミングについて説明するに、ビデオプレゼン
 テーションユニットの表示は、接続点を通して、ギャップなしに連続でなければ
 ならない。ここで、STC1は、TS1のシステムタイムベースの時間軸（図10
 5ではSTC1と図示されている）とし、STC2は、TS2のシステムタイム
 ベースの時間軸（図97ではSTC2と図示されている。正確には、STC2は、
 TS2の最初のPCRがT-STDに入力した時刻から開始する。）とする。

STC1とSTC2の間のオフセットは、次のように決定される。PTS1endは、

T S 1 の最後のビデオプレゼンテーションユニットに対応する S T C 1 上の P T S であり、PTS2startは、T S 2 の最初のビデオプレゼンテーションユニットに対応する S T C 2 上の P T S であり、Tppは、T S 1 の最後のビデオプレゼンテーションユニットの表示期間とすると、2つのシステムタイムベースの間のオフセットSTC_deltaは、次式により算出される。

$$STC_delta = PTS1end + Tpp - PTS2start$$

オーディオのプレゼンテーションのタイミングについて説明するに、接続点において、オーディオプレゼンテーションユニットの表示タイミングのオーバーラップがあってもよく、それは0乃至2オーディオフレーム未満である（図105に図示されている”audio overlap”を参照）。どちらのオーディオサンプルを選択するかということと、オーディオプレゼンテーションユニットの表示を接続点の後の補正されたタイムベースに再同期することは、プレーヤ側により設定されることである。

D V D - S T D のシステムタイムクロックについて説明するに、時刻T5において、T S 1 の最後のオーディオプレゼンテーションユニットが表示される。システムタイムクロックは、時刻T2からT5の間にオーバーラップしていてもよい。この区間では、D V R - S T D は、システムタイムクロックを古いタイムベースの値（S T C 1）と新しいタイムベースの値（S T C 2）の間で切り換える。S T C 2 の値は、次式により算出される。

$$STC2 = STC1 - STC_delta$$

バッファリングの連続性について説明する。STC11video_endは、T S 1 の最後のビデオパケットの最後のバイトがD V R - S T D のT B 1 へ到着する時のシステムタイムベースS T C 1 上のS T C の値である。STC22video_startは、T S 2 の最初のビデオパケットの最初のバイトがD V R - S T D のT B 1 へ到着する時のシステムタイムベースS T C 2 上のS T C の値である。STC21video_endは、STC11video_end の値をシステムタイムベースS T C 2 上の値に換算した値である。STC21video_endは、次式により算出される。

$$STC21video_end = STC11video_end - STC_delta$$

D V R - S T D に従うために、次の2つの条件を満たすことが要求される。先

ず、TS 2の最初のビデオパケットのTB 1への到着タイミングは、次に示す不等式を満たさなければならない。そして、次に示す不等式を満たさなければならない。

$$STC22video_start > STC21video_end + \Delta T1$$

この不等式が満たされるように、Clip1及び／又は、Clip2の部分的なストリームを再エンコード及び／又は、再多重化する必要がある場合は、その必要に応じて行われる。

次に、STC 1とSTC 2を同じ時間軸上に換算したシステムタイムベースの時間軸上において、TS 1からのビデオパケットの入力とそれに続くTS 2からのビデオパケットの入力は、ビデオバッファをオーバーフロー及びアンダーフローさせてはならない。

このようなシンタクス、データ構造、規則に基づくことにより、記録媒体に記録されているデータの内容、再生情報などを適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。

なお、本例は、多重化ストリームとしてMPEG-2トランスポートストリームを例にして説明しているが、これに限らず、MPEG-2プログラムストリームや米国のDirecTVサービス（商標）で使用されているDSSトランスポートストリームについても適用することが可能である。

次に、mark_entry()及びrepresentative_picture_entry()のシンタクスが、図81に示されるような構成である場合における、マーク点で示されるシーンの頭出し再生を行う場合の処理について、図106のフローチャートを参照して、説明する。

最初にステップS1において、記録再生装置1の制御部23は、記録媒体100から、DVRトランスポートストリームファイルのデータベースであるEP_Map（図70）、STC_Info（図52）、Program_Info（図54）、及びClipMark（図78）を読み出す。

ステップS2において、制御部23は、ClipMark（図78）のrepresentative_picture_entry（図81）、又はref_thumbnail_indexで参照されるピクチャから

サムネイルのリストを作成し、ユーザインタフェース入出力としての端子24から出力し、GUIのメニュー画面上に表示させる。この場合、ref_thumbnail_indexが有効な値を持つ場合、representative_picture_entryよりref_thumbnail_indexが優先される。

ステップS3において、ユーザが再生開始点のマーク点を指定する。これは、例えば、GUIとして表示されたメニュー画面上の中からユーザがサムネイル画像を選択することで行われる。制御部23は、この選択操作に対応して、指定されたサムネイルに対応付けられているマーク点を取得する。

ステップS4において、制御部23は、ステップS3で指定されたmark_entry(図81)のmark_time_stampのPTSと、STC_sequence_idを取得する。

ステップS5において、制御部23は、STC_Info(図52)から、ステップS4で取得したSTC_sequence_idに対応するSTC時間軸が開始するソースパケット番号を取得する。

ステップS6において、制御部23は、ステップS5で取得したSTC時間軸が開始するパケット番号と、ステップS4で取得したマーク点のPTSから、マーク点のPTSより時間的に前で、且つ、最も近いエントリポイント(Iピクチャ)のあるソースパケット番号を取得する。

ステップS7において、制御部23は、ステップS6で取得したエントリポイントのあるソースパケット番号から、トランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

ステップS8において、制御部23は、AVデコーダ27を制御し、ステップS4で取得したマーク点のPTSのピクチャから表示を開始させる。

以上の動作を、図107乃至109を参照して更に説明する。

今、図107に示されているように、DVRトランスポートストリームファイルは、STC_sequence_id=id0のSTC時間軸を有し、その時間軸が開始するソースパケット番号は、シーン開始点Aのソースパケット番号より小さいものとする。そして、ソースパケット番号BからCまでの間に、CM(コマーシャル)が挿入されているものとする。

このとき、図70に示されるEP_Mapに対応するEP_Mapには、図108に示され

るように、RSPN_EP_startで示されるA, B, Cに対応して、それぞれのPTSが、PTS_EP_startとして、PTS(A), PTS(B), PTS(C)として登録される。

また、図109に示されるように、図78のClipMarkに対応するClipMarkには、図109に示されるように、シーンスタート、CMスタート、及びCMエンドを表すマークタイプ(図79) 0x92, 0x94, 0x95の値に対応して、mark_entryとrepresentative_picture_entryが記録される。

mark_entryのMark_Time_stampとしては、シーンスタート、CMスタート、及びCMエンドに対応して、それぞれPTS(a1), PTS(b0), PTS(c0)が登録されており、それぞれのSTC_sequence_idは、いずれもid0とされている。

同様に、Representative_picture_entryのMark_Time_stampとして、シーンスタート、CMスタート、及びCMエンドに対応して、それぞれPTS(a2), PTS(b0), PTS(c0)が登録されており、それらはいずれもSTC_sequence_idが、id0とされている。

PTS(A) < PTS(a1)の場合、ステップS6において、パケット番号Aが取得され、ステップS7において、パケット番号Aから始まるトランスポートストリームが、AVデコーダ27に供給され、ステップS8において、PTS(a1)のピクチャから表示が開始される。

次に、図110のフローチャートを参照して、mark_entryとrepresentative_picture_entryのシンタクスが、図81に示されるような構成である場合におけるCMスキップ再生の処理について、図110のフローチャートを参照して説明する。

ステップS21において、制御部23は、EP_map(図70)、STC_Info(図52)、Program_Info(図54)、及びClipMark(図78)を記録媒体100から読み出す。ステップS22において、ユーザは、ユーザインタフェース入出力としての端子24からCMスキップ再生を指定する。

ステップS23において、制御部23は、マークタイプ(図79)がCM開始点(0x94)であるマーク情報のPTSと、CM終了点(0x95)であるマーク情報のPTS、並びに対応するSTC_sequence_idを取得する(図81)。

ステップS 2 4において、制御部 2 3は、STC_Info (図 5 2) からCM開始点と終了点の、STC_sequence_idに対応するS T C時間軸が開始するソースパケット番号を取得する。

ステップS 2 5において、制御部 2 3は、記録媒体 1 0 0からトランスポートストリームを読み出させ、それをA Vデコーダ 2 7に供給し、デコードを開始させる。

ステップS 2 6において、制御部 2 3は、現在の表示画像がCM開始点のP T Sの画像か否かを調べる。現在の表示画像がCM開始点のP T Sの画像でない場合には、ステップS 2 7に進み、制御部 2 3は、画像の表示が継続される。その後、処理はステップS 2 5に戻り、それ以降の処理が繰り返し実行される。

ステップS 2 6において、現在の表示画像がCM開始点のP T Sの画像であると判定された場合、ステップS 2 8に進み、制御部 2 3は、A Vデコーダ 2 7を制御し、デコード及び表示を停止させる。

次に、ステップS 2 9において、制御部 2 3は、CM終了点のSTC_sequence_idに対応するS T C時間軸が開始するパケット番号を取得し、そのパケット番号と、ステップS 2 3の処理で取得したCM終了点のP T Sとから、その点のP T Sより時間的に前で、且つ、最も近いエントリポイントのあるソースパケット番号を取得する。

ステップS 3 0において、制御部 2 3は、ステップS 2 9の処理で取得したエントリポイントのあるソースパケット番号から、トランスポートストリームのデータを読み出し、A Vデコーダ 2 7に供給させる。

ステップS 3 1において、制御部 2 3は、A Vデコーダ 2 7を制御し、CM終了点のP T Sのピクチャから表示を再開させる。

図 1 0 7乃至図 1 0 9を参照して、以上の動作を更に説明すると、CM開始点とCM終了点は、この例の場合、STC_sequence_id=id0という共通のS T C時間軸上に存在し、そのS T C時間軸が開始するソースパケット番号は、シーンの開始点のソースパケット番号Aより小さいものとされている。

トランスポートストリームがデコードされ、ステップS 2 6で、表示時刻がP T S (b 0) になったと判定された場合 (CM開始点であると判定された場合)、

A Vデコーダ27により表示が停止される。そして、 $PTS(C) < PTS(c0)$ の場合、ステップS30でパケット番号Cのデータから始まるストリームからデコードが再開され、ステップS31において、 $PTS(c0)$ のピクチャから表示が再開される。

なお、この方法は、CMスキップ再生に限らず、一般的にClipMarkで指定される2点間のシーンをスキップして再生する場合にも、適用可能である。

次に、mark_entryとrepresentative_picture_entryが、図82に示すシンタクス構造である場合における、マーク点で示されるCMの頭出し再生処理について、図112のフローチャートを参照して説明する。

ステップS41において、制御部23は、EP_map(図70)、STC_Info(図52)、Program_Info(図54)、及びClipMark(図78)の情報を取得する。

次にステップS42において、制御部23は、ステップS41で読み出したClipMark(図78)に含まれるrepresentative_picture_entry(図82)又はref_thumbnail_indexで参照されるピクチャからサムネイルのリストを生成し、GUIのメニュー画面上に表示させる。ref_thumbnail_indexが有効な値を有する場合、representative_picture_entryよりref_thumbnail_indexが優先される。

ステップS43において、ユーザは再生開始点のマーク点を指定する。この指定は、例えば、ステップS42の処理で表示されたメニュー画面上の中から、ユーザがサムネイル画像を選択し、そのサムネイルに対応付けられているマーク点を指定することで行われる。

ステップS44において、制御部23は、ステップS43の処理で指定されたマーク点のRSPN_ref_EP_startとoffset_num_pictures(図82)を取得する。

ステップS45において、制御部23は、ステップS44で取得したRSPN_ref_EP_startに対応するソースパケット番号からトランスポートストリームのデータを読み出し、A Vデコーダ27に供給させる。

ステップS46において、制御部23は、A Vデコーダ27を制御し、RSPN_ref_EP_startで参照されるピクチャから(表示はしないで)、表示すべきピクチャをカウントアップしていき、カウント値がoffset_num_picturesになったとき、そのピクチャから表示を開始させる。

以上の処理を、図 1 1 3 乃至図 1 1 5 を参照して、更に説明する。この例においては、DVR トラストポートストリームファイルは、ソースパケット番号 A からシーンが開始しており、ソースパケット番号 B からソースパケット C まで CM が挿入されている。このため、図 1 1 4 に示されるように、EP_map には、RSPN_EP_start としての A, B, C に対応して、PTS_EP_start として、PTS (A), PTS (B), PTS (C) が登録されている。

また、図 1 1 5 に示されるように、シーンスタート、CM スタート、及び CM エンドのマークタイプに対応して、mark_entry と representative_picture_entry が登録されている。mark_entry には、シーンスタート、CM スタート、及び CM エンドに対応して、RSPN_ref_EP_start として、それぞれ A, B, C が登録され、offset_num_pictures として、M1, N1, N2 が登録されている。同様に、representative_picture_entry には、RSPN_ref_EP_start として、シーンスタート、CM スタート、及び CM エンドに対応して、それぞれ A, B, C が登録され、offset_num_pictures として、M2, N1, N2 がそれぞれ登録されている。

シーンスタートにあたるピクチャから頭出して再生が指令された場合、パケット番号 A のデータから始まるストリームからデコードが開始され、PTS (A) のピクチャから（表示をしないで）表示すべきピクチャをカウントアップをしていき、offset_num_pictures が、M1 の値になったとき、そのピクチャから表示が開始される。

更に、mark_entry と representative_picture_entry のシンタクスが、図 8 2 に示される構成である場合における CM スキップ再生の処理について、図 1 1 6 のフローチャートを参照して説明する。

ステップ S 6 1 において、制御部 2 3 は、EP_map (図 7 0)、STC_Info (図 5 2)、Program_Info (図 5 4)、及び ClipMark (図 7 8) の情報を取得する。

ステップ S 6 2 において、ユーザが CM スキップ再生を指令すると、ステップ S 6 3 において、制御部 2 3 は、マークタイプ (図 7 9) が CM 開始点と CM 終了点である各点のマーク情報として、RSPN_ref_EP_START と offset_num_pictures (図 8 2) を取得する。そして、CM 開始点のデータは、RSPN_ref_EP_start(1), offset_num_pictures(1) とされ、CM 終了点のデータは、RSPN_ref_EP_start(2),

offset_num_pictures(2)とされる。

ステップS 6 4において、制御部 2 3は、RSPN_ref_EP_start(1),RSPN_ref_EP_start(2)に対応するPTSをEP_map (図 7 0) から取得する。

ステップS 6 5において、制御部 2 3は、トランスポートストリームを記録媒体 1 0 0から読み出させ、AVデコーダ 2 7に供給させる。

ステップS 6 6において、制御部 2 3は、現在の表示画像がRSPN_ref_EP_start(1)に対応するPTSのピクチャであるか否かを判定し、現在の表示画像がRSPN_ref_EP_start(1)に対応するPTSのピクチャでない場合には、ステップS 6 7に進み、ピクチャをそのまま継続的に表示させる。その後、処理はステップS 6 5に戻り、それ以降の処理が繰り返し実行される。

ステップS 6 6において、現在の表示画像がRSPN_ref_EP_start(1)に対応するPTSのピクチャであると判定された場合、ステップS 6 8に進み、制御部 2 3は、AVデコーダ 2 7を制御し、RSPN_ref_EP_start(1)に対応するPTSのピクチャから表示するピクチャをカウントアップしていき、カウント値がoffset_num_pictures(1)になったとき、表示を停止させる。

ステップS 6 9において、制御部 2 3は、RSPN_ref_EP_start(2)のソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ 2 7に供給させる。

ステップS 7 0において、制御部 2 3は、AVデコーダ 2 7を制御し、RSPN_ref_EP_start(2)に対応するPTSのピクチャから（表示をしないで）表示すべきピクチャをカウントアップしていき、カウント値がoffset_num_pictures(2)になったとき、そのピクチャから表示を開始させる。

以上の動作を、図 1 1 3乃至図 1 1 5を参照して更に説明すると、まず、EP_map (図 1 1 4) をもとに、パケット番号B, Cに対応する時刻PTS (B), PTS (C) が得られる。そして、Clip AV streamがデコードされていき、表示時刻がPTS (B) になったとき、PTS (B) のピクチャから表示ピクチャがカウントアップされ、その値がN 1 (図 1 1 5) になったとき、表示が停止される。

更に、パケット番号Cのデータから始まるストリームからデコードが再開され、PTS (C) のピクチャから（表示をしないで）表示すべきピクチャをカウント

アップしていき、その値がN 2（図1 1 5）になったとき、そのピクチャから表示が再開される。

以上の処理は、CMスキップ再生に限らず、ClipMarkで指定された2点間のシーンをスキップさせて再生する場合にも、適用可能である。

次に、mark_entryとrepresentative_picture_entryのシンタクスが、図8 4に示すような構成である場合における、マーク点で示されるシーンの頭出し再生処理について、図1 1 8のフローチャートを参照して説明する。

ステップS 8 1において、EP_map（図7 0）、STC_Info（図5 2）、Program_Info（図5 4）、並びにClipMark（図7 8）の情報が取得される。

ステップS 8 2において、制御部2 3は、ClipMark（図7 8）のrepresentative_picture_entry又はref_thumbnail_indexで参照されるピクチャからサムネイルのリストを生成し、GUIのメニュー画面として表示させる。ref_thumbnail_indexが有効な値を有する場合、representative_picture_entryよりref_thumbnail_indexが優先される。

ステップS 8 3において、ユーザは再生開始点のマーク点を指定する。この指定は、例えば、メニュー画面上の中からユーザがサムネイル画像を選択し、そのサムネイルに対応付けられているマーク点を指定することで行われる。

ステップS 8 4において、制御部2 3は、ユーザから指定されたmark_entryのRSPN_mark_point（図8 4）を取得する。

ステップS 8 5において、制御部2 3は、マーク点のRSPN_mark_pointより前にあり、且つ、最も近いエントリポイントのソースバケット番号を、EP_map（図7 0）から取得する。

ステップS 8 6において、制御部2 3は、ステップS 8 5で取得したエントリポイントに対応するソースバケット番号からトランスポートストリームのデータを読み出し、AVデコーダ2 7に供給させる。

ステップS 8 7において、制御部2 3は、AVデコーダ2 7を制御し、RSPN_mark_pointで参照されるピクチャから表示を開始させる。

以上の処理を、図1 1 9乃至図1 2 1を参照して更に説明する。この例においては、DVRトランスポートストリームファイルが、ソースバケットAでシーン

スタートし、ソースバケット番号BからCまでCMが挿入されている。このため、図120のEP_mapには、RSPN_EP_startとしてのA、B、Cに対応して、PTS_EP_startがそれぞれPTS(A)、PTS(B)、PTS(C)として登録されている。また、図121に示されるClipMarkに、シーンスタート、CMスタート、及びCMエンドに対応して、markentryのRSPN_mark_pointとして、a1、b1、c1が、また、representative_picture_entryのRSPN_mark_pointとして、a2、b1、c1が、それぞれ登録されている。

シーンスタートにあたるピクチャから頭出して再生する場合、バケット番号A < a1とすると、バケット番号Aのデータから始まるストリームからデコードが開始され、ソースバケット番号a1に対応するピクチャから表示が開始される。

次に、mark_entryとrepresentative_picture_entryのシンタクスが、図84に示されるような構成である場合におけるCMスキップ再生の処理について、図122と図123のフローチャートを参照して説明する。

ステップS101において、制御部23は、EP_map(図70)、STC_Info(図52)、Program_Info(図54)、並びにClipMark(図70)の情報を取得する。

ステップS102において、ユーザは、CMスキップ再生を指定する。

ステップS103において、制御部23は、マークタイプ(図79)がCM開始点とCM終了点である各点のマーク情報のRSPN_mark_point(図84)を取得する。そして、制御部23は、CM開始点のデータをRSPN_mark_point(1)とし、CM終了点のデータをRSPN_mark_point(2)とする。

ステップS104において、制御部23は、記録媒体100からトランスポートストリームを読み出させ、AVデコーダ27に出力し、デコードさせる。

ステップS105において、制御部23は、現在の表示画像がRSPN_mark_point(1)に対応するピクチャであるか否かを判定し、現在の表示画像がRSPN_mark_point(1)に対応するピクチャでない場合には、ステップS106に進み、そのままピクチャを継続的に表示させる。その後、処理はステップS104に戻り、それ以降の処理が繰り返し実行される。

ステップS105において、現在の表示画像がRSPN_mark_point(1)に対応するピクチャであると判定された場合、ステップS107に進み、制御部23はA

Vデコーダ27を制御し、デコード及び表示を停止させる。

次に、ステップS108において、RSPN_mark_point(2)より前にあり、且つ、最も近いエントリポイントのあるソースパケット番号がEP_map(図70)から取得される。

ステップS109において、制御部23は、ステップS108で取得したエントリポイントに対応するソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

ステップS110において、制御部23は、AVデコーダ27を制御し、RSPN_mark_point(2)で参照されるピクチャから表示を再開させる。

以上の処理を図119乃至図121の例で更に説明すると、Clip AV streamをデコードしていき、ソースパケット番号b1(図121)に対応する表示ピクチャになったとき、表示が停止される。そして、ソースパケット番号C<ソースパケット番号c1とすると、パケット番号Cのデータから始まるストリームからデコードが再開され、ソースパケット番号c1に対応するピクチャになったとき、そのピクチャから表示が再開される。

以上のようにして、図124に示されるように、PlayList上で、タイムスタンプにより所定の位置を指定し、このタイムスタンプを各ClipのClip Informationにおいて、データアドレスに変換し、Clip AV streamの所定の位置にアクセスすることができる。

より具体的には、図125に示されるように、PlayList上において、PlayList Markとしてブックマークやリジューム点を、ユーザが時間軸上のタイムスタンプとして指定すると、そのPlayListは再生するとき、そのPlayListが参照しているClipのClipMarkを使用して、Clip AV streamのシーン開始点やシーン終了点にアクセスすることができる。

なお、ClipMarkのシンタクスは、図78の例に替えて、図126に示すようにすることもできる。

この例においては、RSPN_markが、図78のreserved_for_MakerID, mark_entry()、及びrepresentative_picture_entry()に替えて挿入されている。このRSPN_markの32ビットのフィールドは、AVストリームファイル上で、そのマーク

が参照するアクセスユニットの第1バイト目を含むソースパケットの相対アドレスを示す。RSPN_markは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClip Information fileにおいて定義され、offset_SPNの値を初期値としてカウントされる。

その他の構成は、図7.8における場合と同様である。

ClipMarkのシンタクスは、更に図127に示すように構成することもできる。この例においては、図126におけるRSPN_markの代わりに、RSPN_ref_EP_startとoffset_num_picturesが挿入されている。これらは、図82に示した場合と同様のものである。

図128は、アナログAV信号をエンコードして記録する場合、図81に示したシンタクスのClipMarkの作成について説明するフローチャートである。図1の記録再生装置1のブロック図を参照しながら説明する。ステップS200において、解析部14は端子11, 12からの入力AV信号を解析して、特徴点を検出する。特徴点は、AVストリームの内容に起因する特徴的なシーンを指定し、例えば、番組の頭だし点やシーンチェンジ点などである。

ステップS201において、制御部23は特徴点の画像のPTSを取得する。ステップS202において、制御部23は、特徴点の情報をClipMarkにストアする。具体的には、本例のClipMarkのシンタクスとセマンティクスで説明した情報をストアする。ステップS203において、Clip Information fileとClip AV stream fileがディスクに記録される。

図129は、デジタルインタフェースから入力されたトランスポートストリームを記録する場合、図81に示したシンタクスのClipMarkの作成について説明するフローチャートである。図1の記録再生装置1のブロック図を参照しながら説明する。ステップS211において、デマルチプレクサ26、及び、制御部23は、記録するプログラムのエレメンタリストリームPIDを取得する。解析対象のエレメンタリストリームが複数ある場合、全てのエレメンタリストリームPIDが取得される。

ステップS212で、デマルチプレクサ26は、端子13から入力されるトランスポートストリームのプログラムからエレメンタリストリームを分離し、それ

をA Vデコーダ27がA V信号にデコードする。ステップS 2 1 3において、解析部14は、上記A V信号を解析して特徴点を検出する。

ステップS 2 1 4において、制御部23は、特徴点の画像のP T Sと、それが属するS T CのSTC-sequence-idを取得する。ステップS 2 1 5で、制御部23は、特徴点の情報をClipMarkにストアする。具体的には、本例におけるClipMarkのシタックスとセマンティクスで説明した情報をストアする。

ステップS 2 1 6において、Clip Information fileとClip AV stream fileがディスクに記録される。

図128に示したフローチャート、及び、図129に示したフローチャートのようにして、A Vストリームファイル、すなわちClip A Vストリームファイルの中の特徴的な画像を指し示すマークをストアするClipMarkが、前記A Vストリームの管理情報データファイル、すなわちClip Informationファイルに記録される。

図130は、Real PlayListの作成について説明するフローチャートである。図1の記録再生装置1のブロック図を参照しながら説明する。ステップS 2 2 1において、制御部23はClip A Vストリームを記録する。ステップS 2 2 2において、制御部23は、上記Clipの全ての再生可能範囲をカバーするPlayItemからなるPlayList()を作成する。Clipの中にS T C不連続点があり、PlayList()が2つ以上のPlayItemからなる場合、PlayItem間のconnection_conditionもまた決定される。

ステップS 2 2 3において、制御部23は、UIAppInfoPlayList()を作成する。ステップS 2 2 4において、制御部23は、PlayListMarkを作成する。ステップS 2 2 5において、制御部23は、MakersPrivateDataを作成する。ステップS 2 2 6において、制御部23は、Real PlayListファイルを記録する。

このようにして、新規にClip A Vストリームを記録する毎に、1つのReal PlayListファイルが作られる。

図131は、Virtual PlayListの作成について説明するフローチャートである。ステップS 2 3 1において、ユーザインタフェースを通して、ディスクに記録されている1つのReal PlayListの再生が指定される。そして、そのReal PlayListの再生範囲の中から、ユーザインタフェースを通して、I N点とO U T点で示さ

れる再生区間が指定される。

ステップS 2 3 2において、制御部 2 3は、ユーザによる再生範囲の指定操作が全て終了したか否かを判断する。ステップS 2 3 2において、ユーザによる再生範囲の指定操作はまだ終了していないと判断された場合、ステップS 2 3 1に戻り、それ以降の処理が繰り返され、終了したと判断された場合、ステップS 2 3 3に進む。

ステップS 2 3 3において、連続して再生される2つの再生区間の間の接続状態(connection_condition)が、ユーザがユーザインタフェースを通して決定されるか、又は制御部 2 3により決定される。ステップS 2 3 4において、ユーザインタフェースを通して、ユーザがサブパス(アフレコ用オーディオ)情報を指定する。ユーザがサブパスを作成しない場合、ステップS 2 3 4における処理はスキップされる。

ステップS 2 3 5において、制御部 2 3は、ユーザが指定した再生範囲情報、及びconnection_conditionに基づいて、PlayList()を作成する。ステップS 2 3 6において、制御部 2 3はUIAppInfoPlayList()を作成する。ステップS 2 3 7において、制御部 2 3は、PlayListMarkを作成する。ステップS 2 3 8において、制御部 2 3は、MakersPrivateDataを作成する。ステップS 2 3 9において、制御部 2 3は、Virtual PlayListファイルを、ディスクに記録させる。

このようにして、ディスクに記録されているReal PlayListの再生範囲の中から、ユーザが、見たい再生区間を選択し、その再生区間をグループ化したものに、1つのVirtual PlayListファイルが作成される。

図1 3 2は、PlayListの再生について説明するフローチャートである。図1の記録再生装置1のブロック図を参照しながら説明する。ステップS 2 4 1において、制御部 2 3は、Info.dvr, Clip Information file, PlayList file及びサムネイルファイルの情報を取得し、ディスクに記録されているPlayListの一覧を示すGUI画面を作成し、ユーザインタフェースを通して、GUIに表示する。

ステップS 2 4 2において、ユーザインタフェースを通して、ユーザが1つのPlayListの再生を制御部 2 3に指示する。ステップS 2 4 3において、制御部 2 3は、現在のPlayItemのSTC-sequence-idとIN_timeのPTSから、IN_timeより時

間的に前で最も近いエントリポイントのあるソースパケット番号を取得する。ステップS 2 4 4において、制御部2 3は、上記エントリポイントのあるソースパケット番号からA Vストリームのデータを読み出し、A Vデコーダ2 7へ供給する。

上記PlayItemの時間的に前にPlayItemの再生があった場合、ステップS 2 4 5において、制御部2 3は、そのPlayItemとの表示の接続処理をconnection_conditionに従って行なわれるように制御を行う。ステップS 2 4 6において、A Vデコーダ2 7は、IN_timeのP T Sのピクチャから表示を開始する。

ステップS 2 4 7において、A Vデコーダ2 7は、A Vストリームのデコードを継続的に行う。ステップS 2 4 8において、制御部2 3は、現在表示の画像が、OUT_timeのP T Sの画像か否かを判断する。ステップS 2 4 8において、現在表示の画像は、OUT_timeのP T Sの画像であると判断された場合、ステップS 2 5 0に進み、P T Sの画像ではないと判断された場合、ステップS 2 4 9に進む。

ステップS 2 4 9において、P T Sの画像であると判断された画像を表示するための処理が実行され、その後ステップS 2 4 7に戻り、それ以降の処理が繰り返される。一方、ステップS 2 5 0においては、制御部2 3により、現在のPlayItemがPlayListの中で最後のPlayItemか否かが判断される。ステップS 2 5 0において、現在のPlayItemがPlayListの中で最後のPlayItemであると判断された場合、図1 3 2に示したフローチャートの処理は終了され、最後のPlayItemではないと判断された場合、ステップS 2 4 3に戻り、それ以降の処理が繰り返される。

図1 3 3は、PlayListMarkの作成について説明するフローチャートである。図1の記録再生装置1のブロック図を参照しながら説明する。ステップS 2 6 1において、制御部2 3は、Info.dvr, Clip Information file, PlayList file及びThumbnail fileの情報を取得し、ディスクに記録されているPlayListの一覧を示すG U I画面を作成し、ユーザインタフェースを通して、G U Iに表示する。

ステップS 2 6 2において、ユーザインタフェースを通して、ユーザにより1つのPlayListの再生が制御部2 3に指示される。ステップS 2 6 3において、再生部3は、指示されたPlayListの再生を開始する（図1 3 2のフローチャートを参照して説明したように行われる）。

ステップS 2 6 4において、ユーザインタフェースを通して、ユーザにより、お気に入りのシーンのところにマークのセットが制御部 2 3 に指示される。ステップS 2 6 5において、制御部 2 3 は、マークのPTSと、それが属するPlayItemのPlayItem_idを取得する。

ステップS 2 6 6において、制御部 2 3 は、マークの情報をPlayListMark()にストアする。ステップS 2 6 7において、PlayListファイルがディスクに記録される。

このようにして、PlayListの再生範囲の中からユーザが指定したマーク点、又は、そのPlayListを再生するときのResume点を示すマークをストアするPlayListMarkを、PlayListファイルに記録される。

図 1 3 4 は、PlayListが再生される時、PlayListMark及びそのPlayListが参照するClipのClipMarkが使用された頭だし再生について説明するフローチャートである。ClipMark()のシンタクスは、図 8 1 に示すものとする。図 1 の記録再生装置 1 のブロック図を参照しながら説明する。

ステップS 2 7 1において、制御部 2 3 は、Info.dvr, Clip Information file, PlayList file及びThumbnail fileの情報を取得し、ディスクに記録されているPlayListの一覧を示すGUI画面を作成し、ユーザインタフェースを通して、GUIに表示する。

ステップS 2 7 2において、ユーザインタフェースを通して、ユーザにより1つのPlayListの再生が指示される。ステップS 2 7 3において、制御部 2 3 は、PlayListMark、及び、そのPlayListが参照するClipのClipMarkで参照されるピクチャから生成したサムネイルのリストを、ユーザインタフェースを通して、GUIに表示する。

ステップS 2 7 4において、ユーザインタフェースを通して、制御部 2 3 に、ユーザにより再生開始点のマーク点が指定される。ステップS 2 7 5において、制御部 2 3 は、ステップS 2 7 4における処理で選択されたマークがPlayListMarkにストアされているマークか否かを判断する。ステップS 2 7 5において、選択されたマークがPlayListMarkにストアされているマークであると判断された場合、ステップS 2 7 6に進み、ストアされていないマークであると判断された場

合、ステップS 2 7 8に進む。

ステップS 2 7 6において、制御部23は、マークのPTSと、それが属するPlayItem_idを取得する。ステップS 2 7 7において、制御部23はPlayItem_idが指すPlayItemが参照するAVストリームのSTC-sequence-idを取得する。

ステップS 2 7 8において、制御部23は、STC-sequence-idとマークのPTSに基づいて、AVストリームをAVデコーダ27へ入力させる。具体的には、このSTC-sequence-idとマーク点のPTSを用いて、図13-2のフローチャートのステップS 2 4 3, S 2 4 4と同様の処理が行なわれる。ステップS 2 7 9において、再生部3は、マーク点のPTSのピクチャから表示を開始する。

図9を参照して説明したように、Playlistが再生される時、そのPlaylistが参照するClipのClipMarkにストアされているマークを参照することができる。したがって、1つのClipを、Real Playlistや複数のVirtual Playlistによって参照している場合、それらのPlaylistは、その1つのClipのClipMarkを共有することができるので、マークのデータを効率良く管理することができる。

仮に、ClipにClipMarkを定義しないで、PlaylistだけにPlaylistMarkとClipMarkを合わせたものを定義するようにした場合、上記の例のように1つのClipをReal Playlistや複数のVirtual Playlistによって参照している場合、それぞれのPlaylistが同じ内容のClipのマーク情報を持つことになり、データの記録の効率が悪い。

図135は、PlaylistMark()のシンタクスの別例を示す図である。lengthは、このlengthフィールドの直後のバイトからPlaylistMark()の最後のバイトまでのバイト数を示す。number_of_PlayList_marksは、PlaylistMarkの中にストアされているマークのエントリ数を示す。

mark_invalid_flagは、1ビットのフラグであり、この値が0にセットされている時、このマークは有効な情報を持っていることを示し、また、この値が1にセットされている時、このマークは無効であることを示す。

ユーザがユーザインタフェース上で1つのマークのエントリを消去するオペレーションをした時、記録再生装置1は、PlaylistMarkからそのマークのエントリを消去する代わりに、そのmark_invalid_flagの値を1に変更するようにしてもよ

い。

mark_typeは、マークのタイプを示し、図 1 3 6 に示す意味を持つ。mark_name_lengthは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。このフィールドの値は 3 2 以下である。ref_to_PlayItem_idは、マークが置かれているところのPlayItemを指定するところのPlayItem_idの値を示す。あるPlayItemに対応するPlayItem_idの値は、PlayList()において定義される。

mark_time_stampは、そのマークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampは、ref_to_PlayItem_idで示されるPlayItemの中で定義されているところのIN_timeとOUT_timeで特定される再生範囲の中の時間を指す。タイムスタンプの意味は、図 4 4 と同じである。

entry_ES_PIDが、0xFFFFにセットされている場合、そのマークはPlayListによって使用される全てのエレメンタリーストリームに共通の時間軸上へのポイントである。entry_ES_PIDが、0xFFFFでない値にセットされている場合、entry_ES_PIDは、そのマークによって指されるところのエレメンタリーストリームを含んでいるところのトランスポートバケットのP I Dの値を示す。

ref_thumbnail_indexは、マークに付加されるサムネイル画像の情報を示す。その意味は、図 4 2 のref_thumbnail_indexと同じである。mark_nameは、マークの名前を示す。このフィールドの中の左からmark_name_lengthで示されるバイト数が、有効なキャラクター文字であり、名前を示す。このキャラクター文字は、UI AppInfoPlayListの中でcharacter_setによって示される方法で符号化されている。

mark_nameフィールドの中で、それら有効なキャラクター文字に続くバイトの値は、どんな値が入っていてもよい。このシンタクスの場合、マークが特定のエレメンタリーストリームを指すことができる。例えば、PlayListが、プログラムの中に複数のビデオストリームを持つマルチビュープログラムを参照している時、entry_ES_PIDは、そのプログラムの中の 1 つのビデオストリームを示すビデオP I Dをセットするために使われる。

ユーザがマルチビュープログラムを参照するところのPlayListを再生しており、そのユーザは、マルチビュー中の 1 つのビューを見ているとする。今、ユーザが記録再生装置 1 に対して、次のマーク点に再生をスキップするようにコマンドを

送ったとする。この場合、記録再生装置 1 は、ユーザが現在見ているビューのビデオ P I D と同じ値であるところの entry_ES_PID のマークを使用すべきであり、記録再生装置 1 は、勝手にビューを変更すべきでない。記録再生装置 1 は、また、entry_ES_PID が 0xFFFF にセットされているマークを使用してもよい。この場合も記録再生装置 1 は、勝手にビューを変更しない。

図 1 3 7 は、図 8 1 に示すシンタクスの ClipMark() の別例を示す図である。length は、この length フィールドの直後のバイトから ClipMark() の最後のバイトまでのバイト数を示す。maker_ID は、mark_type が 0x60 から 0x7F の値を示す時に、その mark_type を定義しているメーカーのメーカー I D を示す。

number_of_clip_marks は、ClipMark の中にストアされているマークのエントリ数を示す。mark_invalid_flag は、1 ビットのフラグであり、この値が 0 にセットされている時、このマークは有効な情報を持っていることを示し、また、この値が 1 にセットされている時、このマークは無効であることを示す。

ユーザが、ユーザインタフェース上で 1 つのマークのエントリを消去するオペレーションをした時、記録機は ClipMark からそのマークのエントリを消去する代わりに、その mark_invalid_flag の値が 1 に変更されるようにしてもよい。mark_type は、マークのタイプを示し、図 1 3 8 に示す意味を持つ。

ref_to_STC_id は、mark_time_stamp と representative_picture_time_stamp の両方が置かれているところの STC-sequence を指定するところの STC-sequence-id を示す。STC-sequence-id の値は、STCInfo() の中で定義される。mark_time_stamp は、図 8 1 の mark_entry() の場合での mark_time_stamp と同じ意味である。

entry_ES_PID が、0xFFFF にセットされている場合、そのマークは Clip 中の全てのエレメンタリーストリームに共通の時間軸上へのポイントである。entry_ES_PID が、0xFFFF でない値にセットされている場合、entry_ES_PID は、そのマークによって指されるところのエレメンタリーストリームを含んでいるところのトランスポートパケットの P I D の値を示す。

ref_to_thumbnail_index は、マークに付加されるサムネイル画像の情報を示す。その意味は、図 7 8 の ref_thumbnail_index と同じである。representative_picture_time_stamp は、図 8 1 の representative_picture_entry() の場合での mark_t

ime_stampと同じ意味である。

図137に示したシンタクスの場合、マークが、特定のエレメンタリーストリームを指すことができる。例えば、Clipが、プログラムの中に複数のビデオストリームを持つマルチビュープログラムを含んでいるとき、entry_ES_PIDは、そのプログラムの中の1つのビデオストリームを示すビデオPIDをセットするために使われる。

ユーザが、マルチビュープログラムを参照するところのPlayListを再生しており、そのユーザは、マルチビュー中の1つのビューを見ているとする。今、ユーザが記録再生装置1に対して、次のマーク点に再生をスキップするようにコマンドを送ったとする。この場合、記録再生装置1は、ユーザが現在見ているビューのビデオPIDと同じ値であるところのentry_ES_PIDのマークを使用すべきであり、記録再生装置1は、勝手にビューを変更すべきでない。記録再生装置1は、また、entry_ES_PIDが0xFFFFにセットされているマークを使用してもよい。この場合も記録再生装置1は、勝手にビューを変更しない。

このようなシンタクス、データ構造、規則に基づくことにより、記録媒体100に記録されているデータの内容、再生情報などを適切に管理することができ、もって、ユーザが、再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。

以上のようなデータベース構成によれば、PlayListファイルやClip Informationファイルを別々に分離して記録するので、編集などによって、所定のPlayListやClipの内容が変更されたとき、そのファイルに関係のない他のファイルを変更する必要がない。したがって、ファイルの内容の変更が容易に行え、またその変更及び記録にかかる時間を小さくできる。

また、最初にInfo.dvrだけを読み出して、ディスクの記録内容をユーザインタフェースへ提示し、ユーザが再生指示したPlayListファイルと、それに関連するClip Informationファイルだけをディスクから読み出すようにすれば、ユーザの待ち時間を小さくすることができる。

仮に、全てのPlayListファイルやClip Informationファイルを1つのファイルにまとめて記録すると、そのファイルサイズは非常に大きくなる。そのために、

そのファイルの内容を変更して、それを記録するためにかかる時間は、個々のファイルを別々に分離して記録する場合に比べて、非常に大きくなる。本発明を適用することにより、このようなことを防ぐことが可能となる。

上述したように、A Vストリームファイル、すなわちClip A Vストリームファイルの中の特徴的な画像を指し示すマークをストアするClipMarkを、前記A Vストリームの管理情報データファイル、すなわちClip Informationファイルに記録し、また、A Vストリーム中の指定された区間の組み合わせにより定義される1つの再生手順の情報を持つオブジェクト、すなわちPlayListの再生範囲の中から、ユーザが指定したマーク点、又は、そのオブジェクトを再生するときのResume点を示すマークをストアするPlayListMarkを、オブジェクトに記録する。

このようにすることにより、PlayListが再生される時、そのPlayListが参照するClipのClipMarkにストアされているマークを参照することができる。したがって、1つのClipをReal PlayListや複数のVirtual PlayListによって参照している場合、それらのPlayListは、その1つのClipのClipMarkを共有することができるので、マークのデータを効率良く管理することができる。

仮に、ClipにClipMarkを定義しないで、PlayListだけにPlayListMarkとClipMarkを合わせたものを定義するようにした場合、上記の例のように1つのClipをReal PlayListや複数のVirtual PlayListによって参照している場合、それぞれのPlayListが同じ内容のClipのマーク情報を持つことになり、データの記録の効率が悪い。本発明を適用することにより、このようなことを防ぐことが可能となる。

以上のように、A Vストリームの付属情報として、エントリポイントのアドレスをストアするためのEP_mapと、マーク点のピクチャのタイプ（例えば番組の頭出し点）とそのピクチャのA Vストリームの中のアドレスをストアするためのClipMarkを、Clip Information Fileとしてファイル化して記録媒体100に記録することにより、A Vストリームの再生に必要なストリームの再生に必要なストリームの符号化情報を適切に管理することが可能である。

このClip Information file情報により、ユーザが、記録媒体100に記録されているA Vストリームの中から興味のあるシーン、例えば番組の頭出し点など、をサーチすることができ、ユーザのランダムアクセスや特殊再生の指示に対して、

記録媒体 100 からの AV ストリームの読み出し位置の決定が容易になり、またストリームの復号開始を速やかに行うことができる。

上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。この場合、例えば、記録再生装置 1 は、図 139 に示されるようなパーソナルコンピュータにより構成される。

図 139 において、CPU (Central Processing Unit) 201 は、ROM (Read Only Memory) 202 に記憶されているプログラム、又は記憶部 208 から RAM (Random Access Memory) 203 にロードされたプログラムに従って各種の処理を実行する。RAM 203 には、CPC 201 が各種の処理を実行する上において必要なデータなども適宜記憶される。

CPC 201、ROM 202、及び RAM 203 は、バス 204 を介して相互に接続されている。このバス 204 にはまた、入出力インタフェース 205 も接続されている。

入出力インタフェース 205 には、キーボード、マウスなどよりなる入力部 206、CRT、LCD などよりなるディスプレイ、並びにスピーカなどよりなる出力部 207、ハードディスクなどより構成される記憶部 208、モデム、ターミナルアダプタなどより構成される通信部 209 が接続されている。通信部 209 は、ネットワークを介しての通信処理を行う。

入出力インタフェース 205 にはまた、必要に応じてドライブ 210 が接続され、磁気ディスク 221、光ディスク 222、光磁気ディスク 223、或いは半導体メモリ 224 などが適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じて記憶部 208 にインストールされる。

上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが専用のハードウェアに組み込まれているコンピュータ、又は、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどに、記録媒体からインストールされる。

この記録媒体は、図 139 に示すように、コンピュータとは別に、ユーザにプ

プログラムを提供するために配布される、プログラムが記録されている磁気ディスク 2 2 1 (フロッピディスクを含む)、光ディスク 2 2 2 (CD-ROM (Compact Disk-Read Only Memory) , DVD (Digital Versatile Disk) を含む)、光磁気ディスク 2 2 3 (MD (Mini-Disk) を含む)、若しくは半導体メモリ 2 2 4 などよりなるパッケージメディアにより構成されるだけでなく、コンピュータに予め組み込まれた状態でユーザに提供される、プログラムが記憶されている ROM 2 0 2 や記憶部 2 0 8 が含まれるハードディスクなどで構成される。

なお、本明細書において、媒体により提供されるプログラムを記述するステップは、記載された順序に従って、時系列的に行われる処理は勿論、必ずしも時系列的に処理されなくとも、並列的或いは個別に実行される処理をも含むものである。

また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

産業上の利用可能性

以上の如く本発明に係る情報処理装置及び方法、並びにプログラムにおいては、入力された AV ストリームから抽出された特徴的な画像を指し示すマークで構成される ClipMark を、AV ストリームを管理するための管理情報として生成すると共に、AV ストリーム中の所定の区間の組み合わせを定義する PlayList に対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成される PlayListMark を生成し、ClipMark、及び PlayListMark を各々独立したテーブルとして記録媒体に記録するようにしたので、AV ストリームの所望の位置に、迅速且つ確実にアクセスすることが可能となる。

また、本発明に係る情報処理装置及び方法、並びにプログラムは、AV ストリームから抽出された特徴的な画像を指し示すマークで構成される ClipMark を含む AV ストリームを管理するための管理情報と、AV ストリーム中の所定の区間の組み合わせを定義する PlayList に対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成される PlayListMark を読み出し、その読み

出された管理情報とPlayLisMarkによる情報を提示し、提示された情報から、ユーザが再生を指示したPlayListに対応するClipMarkを参照し、参照されたClipMarkを含み、ClipMarkに対応する位置からA Vストリームを再生するようにしたので、A Vストリームの所望の位置に、迅速且つ確実にアクセスすることが可能となる。

請求の範囲

1. 入力されたA Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、前記A Vストリームを管理するための管理情報として生成すると共に、

前記A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成手段と、

前記ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する記録手段とを有する情報処理装置。

2. 前記生成手段は、前記ClipMarkをClipMarkInformationファイルとして生成すると共に、前記PlayListをPlayListファイルとして生成する請求の範囲第1項に記載の情報処理装置。

3. 前記PlayListMarkは、前記PlayListを再生するときのResume点を示すマークを更に含む特徴とする請求の範囲第1項に記載の情報処理装置。

4. 前記PlayListを再生するとき、前記PlayListの再生区間に対応する前記A VストリームのClipMarkを構成する前記マークを参照する請求の範囲第1項に記載の情報処理装置。

5. 前記PlayListMarkの前記マークは、プレゼンテーションタイムスタンプと、前記PlayListの再生経路を構成する前記A Vストリームデータ上の指定された1つの再生区間を示す識別情報を含む請求の範囲第1項に記載の情報処理装置。

6. 前記ClipMarkを構成する前記マーク、又は、前記PlayListMarkを構成する前記マークは、エレメンタリーストリームのエントリポイントを特定する情報を含む請求の範囲第1項に記載の情報処理装置。

7. 前記PlayListMarkの前記マークは、ユーザが指定したお気に入りのシーンの開始点又はPlayListのResume点を少なくとも含むタイプの情報を含む請求の範囲第1項に記載の情報処理装置。

8. 前記ClipMarkを構成する前記マークと前記PlayListMarkを構成する前記マークは、前記A Vストリームのエントリポイントに対応する相対的なソースパ

ケットのアドレスで表される請求の範囲第1項に記載の情報処理装置。

9. 前記ClipMarkを構成する前記マークと前記PlayListMarkを構成する前記マークは、前記AVストリームのエントリポイントに対応する相対的なソースバケットの第1のアドレスと、前記第1のアドレスからのオフセットのアドレスである第2のアドレスで表される請求の範囲第8項に記載の情報処理装置。

10. 前記第1の記録手段による記録の際に検出された前記特徴的な画像のタイプを検出するタイプ検出手段を更に含み、

前記第1の記録手段は、前記ClipMarkを構成する前記マークと、前記タイプ検出手段により検出された前記タイプとを対応させて記録する請求の範囲第1項に記載の情報処理装置。

11. 前記ClipMarkの前記マークは、シーンチェンジ点、コマーシャルの開始点、コマーシャルの終了点、又はタイトルが表示されたシーンを含む請求の範囲第1項に記載の情報処理装置。

12. 入力されたAVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、前記AVストリームを管理するための管理情報として生成すると共に、前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成ステップと、

前記ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御ステップとを有する情報処理方法。

13. 入力されたAVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、前記AVストリームを管理するための管理情報として生成すると共に、前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成ステップと、

前記ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御ステップとを含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

14. 入力されたAVストリームから抽出された特徴的な画像を指し示すマ

ークで構成されるClipMarkを、前記A Vストリームを管理するための管理情報として生成すると共に、前記A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成ステップと、

前記ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御ステップとをコンピュータに実行させるプログラム。

15. A Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含む前記A Vストリームを管理するための管理情報と、前記A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを読み出す読出手段と、

前記読出手段により読み出された前記管理情報と前記PlayListMarkによる情報を提示する提示手段と、

前記提示手段により提示された前記情報から、ユーザが再生を指示した前記PlayListに対応する前記ClipMarkを参照する参照手段と、

前記参照手段により参照された前記ClipMarkを含み、前記ClipMarkに対応する位置から前記A Vストリームを再生する再生手段とを含む情報処理装置。

16. 前記提示手段は、前記PlayListMarkに対応するサムネイル画像によるリストをユーザに提示する請求の範囲第15項に記載の情報処理装置。

17. 前記ClipMarkを構成する前記マークと前記PlayListMarkを構成する前記マークは、前記A Vストリームのエントリポイントに対応する相対的なソースバケットのアドレスで表される請求の範囲第15項に記載の情報処理装置。

18. 前記ClipMarkを構成する前記マークと前記PlayListMarkを構成する前記マークは、前記A Vストリームのエントリポイントに対応する相対的なソースバケットの第1のアドレスと、前記第1のアドレスからのオフセットのアドレスである第2のアドレスで表される請求の範囲第17項に記載の情報処理装置。

19. 前記ClipMarkの前記マークは、シーンチェンジ点、コマーシャルの開始点、コマーシャルの終了点、又はタイトルが表示されたシーンを含む請求の範

図第 15 項に記載の情報処理装置。

20. AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含む前記AVストリームを管理するための管理情報と、前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkの読み出しを制御する読出制御ステップと、

前記読出制御ステップの処理で読み出しが制御された前記管理情報と前記PlayListMarkによる情報を提示する提示ステップと、

前記提示ステップの処理で提示された前記情報から、ユーザが再生を指示した前記PlayListに対応する前記ClipMarkを参照する参照ステップと、

前記参照ステップの処理で参照された前記ClipMarkを含み、前記ClipMarkに対応する位置からの前記AVストリームの再生を制御する再生制御ステップとを含む情報処理方法。

21. AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含む前記AVストリームを管理するための管理情報と、前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkの読み出しを制御する読出制御ステップと、

前記読出制御ステップの処理で読み出しが制御された前記管理情報と前記PlayListMarkによる情報を提示する提示ステップと、

前記提示ステップの処理で提示された前記情報から、ユーザが再生を指示した前記PlayListに対応する前記ClipMarkを参照する参照ステップと、

前記参照ステップの処理で参照された前記ClipMarkを含み、前記ClipMarkに対応する位置からの前記AVストリームの再生を制御する再生制御ステップとを含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

22. AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含む前記AVストリームを管理するための管理情報と、前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlay

ListMarkの読み出しを制御する読出制御ステップと、

前記読出制御ステップの処理で読み出しが制御された前記管理情報と前記PlayListMarkによる情報を提示する提示ステップと、

前記提示ステップの処理で提示された前記情報から、ユーザが再生を指示した前記PlayListに対応する前記ClipMarkを参照する参照ステップと、

前記参照ステップの処理で参照された前記ClipMarkを含み、前記ClipMarkに対応する位置からの前記AVストリームの再生を制御する再生制御ステップとをコンピュータに実行させるプログラム。

23. AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含む前記AVストリームを管理するための管理情報と、前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが、各々独立したテーブルとして記録されている記録媒体。

補正書の請求の範囲

[2001年8月3日(03.08.01)国際事務局受理:出願当初の請求の範囲3, 15, 16及び20-23は補正された;他の請求の範囲は変更なし。(4頁)]

1. 入力されたAVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、前記AVストリームを管理するための管理情報として生成すると共に、

前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成手段と、

前記ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する記録手段とを有する情報処理装置。

2. 前記生成手段は、前記ClipMarkをClipMarkInformationファイルとして生成すると共に、前記PlayListをPlayListファイルとして生成する請求の範囲第1項に記載の情報処理装置。

3. (補正後)前記PlayListMarkは、前記PlayListを再生するときのResume点を示すマークを更に含む請求の範囲第1項に記載の情報処理装置。

4. 前記PlayListを再生するとき、前記PlayListの再生区間に対応する前記AVストリームのClipMarkを構成する前記マークを参照する請求の範囲第1項に記載の情報処理装置。

5. 前記PlayListMarkの前記マークは、プレゼンテーションタイムスタンプと、前記PlayListの再生経路を構成する前記AVストリームデータ上の指定された1つの再生区間を示す識別情報を含む請求の範囲第1項に記載の情報処理装置。

6. 前記ClipMarkを構成する前記マーク、又は、前記PlayListMarkを構成する前記マークは、エレメンタリーストリームのエントリポイントを特定する情報を含む請求の範囲第1項に記載の情報処理装置。

7. 前記PlayListMarkの前記マークは、ユーザが指定したお気に入りのシーンの開始点又はPlayListのResume点を少なくとも含むタイプの情報を含む請求の範囲第1項に記載の情報処理装置。

8. 前記ClipMarkを構成する前記マークと前記PlayListMarkを構成する前記マークは、前記AVストリームのエントリポイントに対応する相対的なソースバ

ークで構成されるClipMarkを、前記A Vストリームを管理するための管理情報として生成すると共に、前記A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成ステップと、

前記ClipMark、及びPlayListMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御ステップとをコンピュータに実行させるプログラム。

15. (補正後) A Vストリーム、A Vストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMark、及び前記A Vストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが記録された記録媒体を再生する情報処理装置であって、

前記記録媒体を再生する再生手段と、

再生された前記ClipMark又はPlayListMarkに記述されたマークに対応する記録位置を取得すると共に、当該取得された記録位置に応じて前記再生手段を制御する制御手段とを含む情報処理装置。

16. (補正後) 更に、前記PlayListMarkに対応するサムネイル画像によるリストをユーザに提示するよう制御する提示制御手段を有する請求の範囲第15項に記載の情報処理装置。

17. 前記ClipMarkを構成する前記マークと前記PlayListMarkを構成する前記マークは、前記A Vストリームのエントリポイントに対応する相対的なソースバケットのアドレスで表される請求の範囲第15項に記載の情報処理装置。

18. 前記ClipMarkを構成する前記マークと前記PlayListMarkを構成する前記マークは、前記A Vストリームのエントリポイントに対応する相対的なソースバケットの第1のアドレスと、前記第1のアドレスからのオフセットのアドレスである第2のアドレスで表される請求の範囲第17項に記載の情報処理装置。

19. 前記ClipMarkの前記マークは、シーンチェンジ点、コマーシャルの開始点、コマーシャルの終了点、又はタイトルが表示されたシーンを含む請求の範囲第15項に記載の情報処理装置。

20. (補正後) AVストリーム、AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMark、及び前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが記録された記録媒体を再生する情報処理方法であって、

前記記録媒体を再生する再生ステップと、

再生された前記ClipMark又はPlayListMarkに記述されたマークに対応する記録位置を取得すると共に、当該取得された記録位置に応じて再生位置を制御する制御ステップとを含む情報処理方法。

21. (補正後) AVストリーム、AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMark、及び前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが記録された記録媒体を再生するためのコンピュータが読み取り可能なプログラムが記録されている記録媒体であって、

前記記録媒体を再生する再生ステップと、

再生された前記ClipMark又はPlayListMarkに記述されたマークに対応する記録位置を取得すると共に、当該取得された記録位置に応じて再生位置を制御する制御ステップと

を含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

22. (補正後) AVストリーム、AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMark、及び前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが記録された記録媒体を再生するためのコンピュータが読み取り可能なプログラムであって、

前記記録媒体を再生する再生ステップと、

再生された前記ClipMark又はPlayListMarkに記述されたマークに対応する記録位置を取得すると共に、当該取得された記録位置に応じて再生位置を制御する制御ステップと

をコンピュータに実行させるプログラム。

23. (補正後) AVストリームが記録されると共に、当該AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMark、及び前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが、各々独立したテーブルとして記録されている記録媒体。

1/118

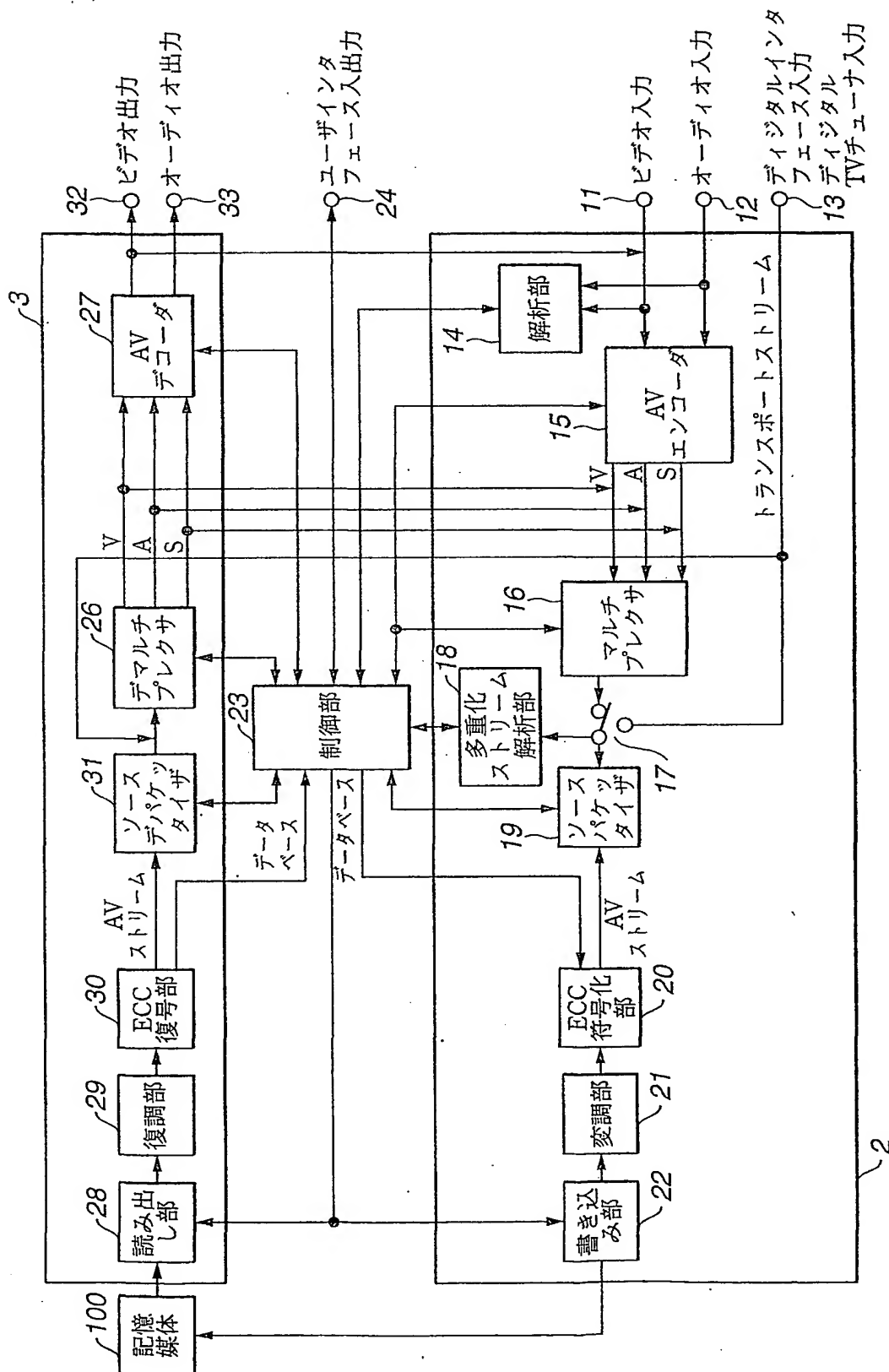


FIG. 1

2/118

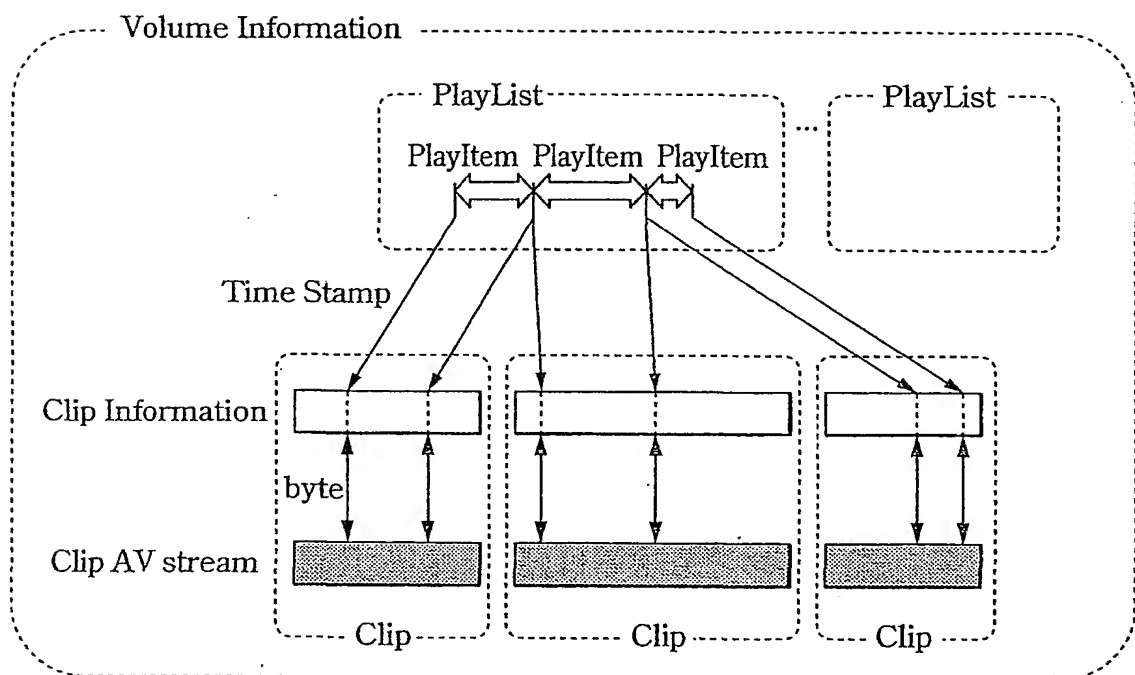


FIG.2

3/118

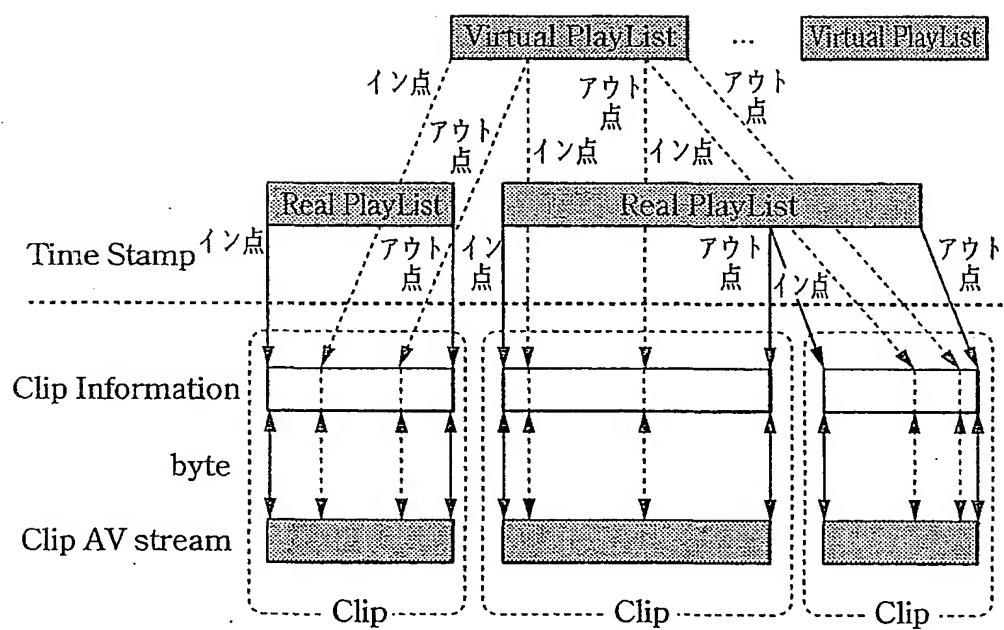


FIG.3

4/118

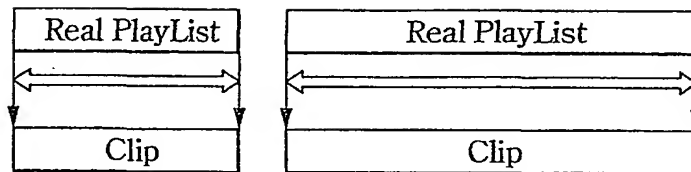


FIG.4A

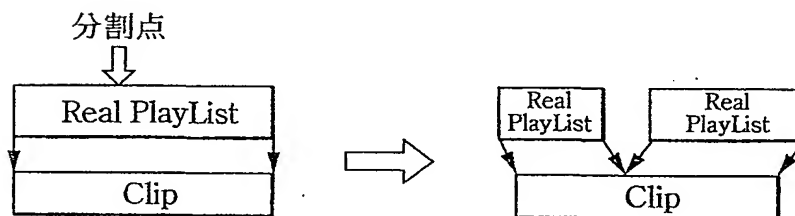


FIG.4B

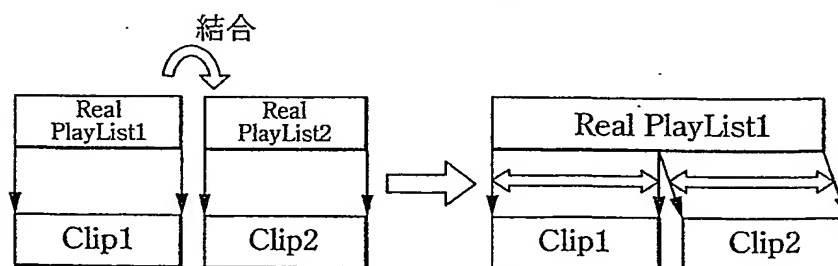


FIG.4C

5/118

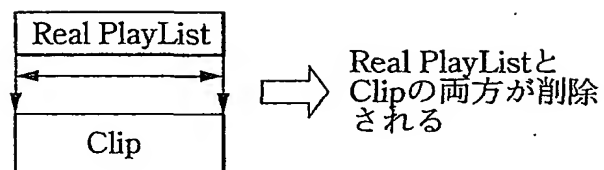


FIG.5A

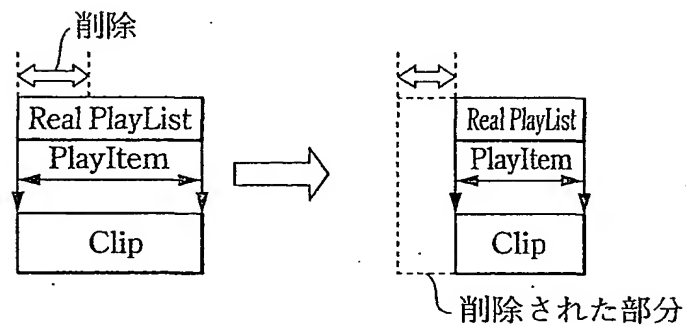


FIG.5B

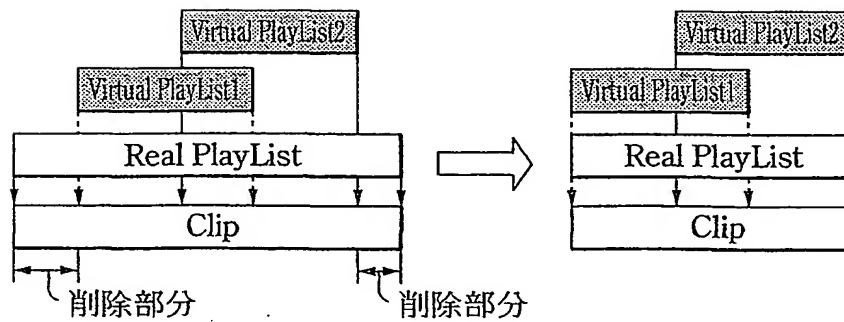
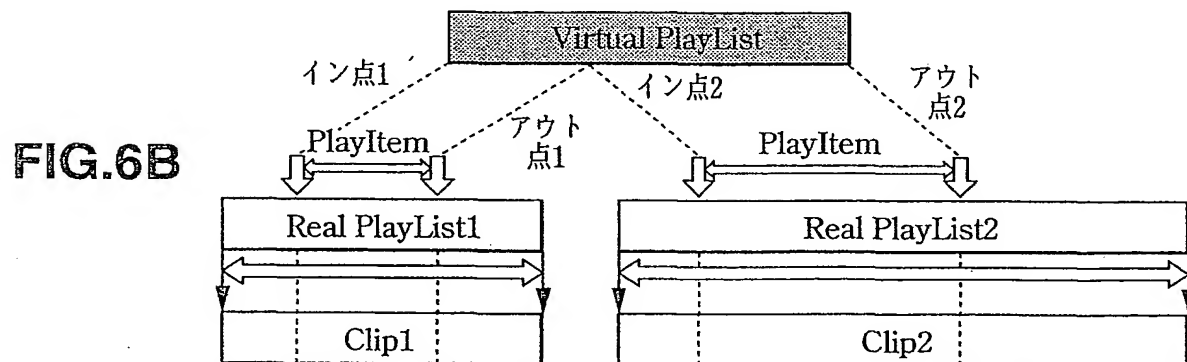
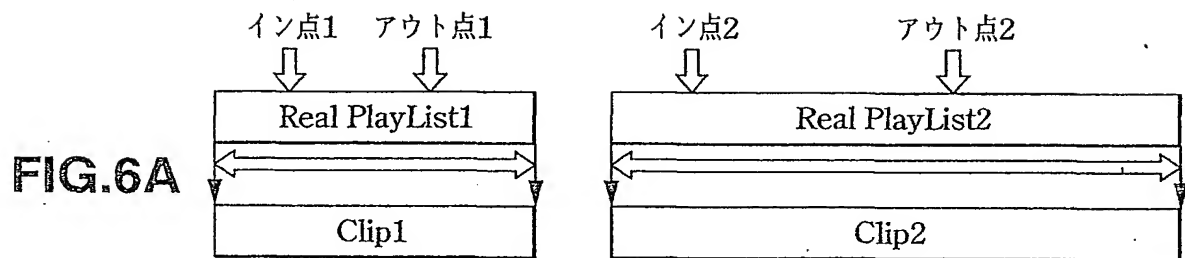


FIG.5C

6/118



7/118

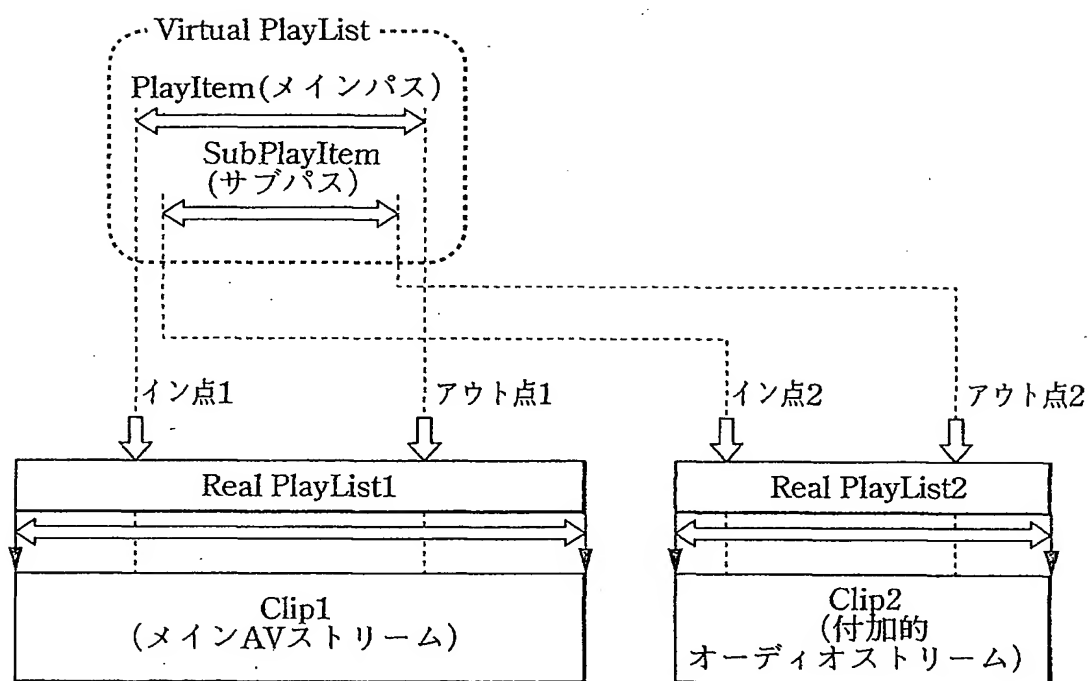


FIG.7

8/118

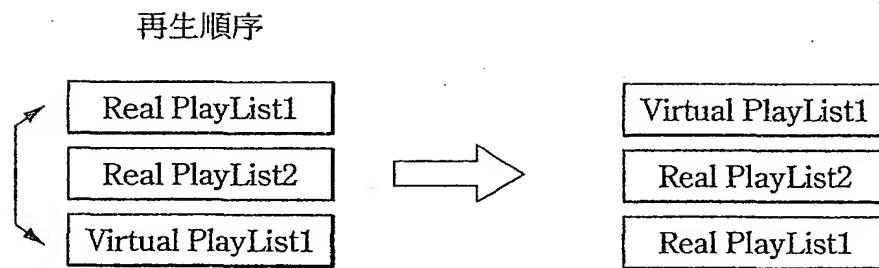


FIG.8

9/118

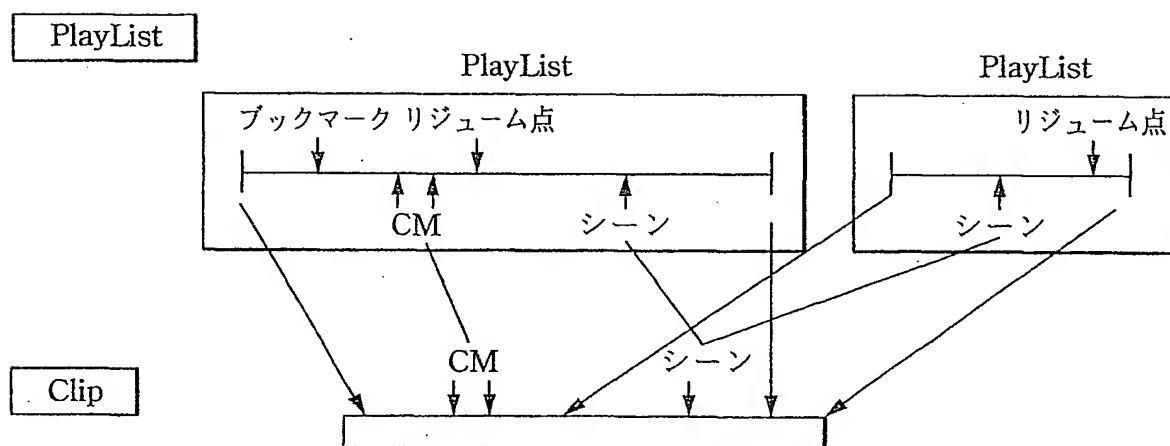


FIG.9

10/118

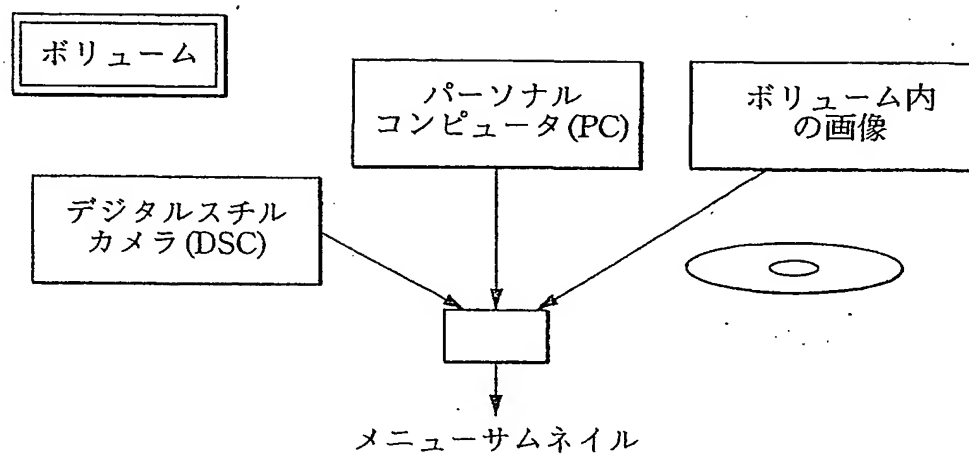


FIG.10

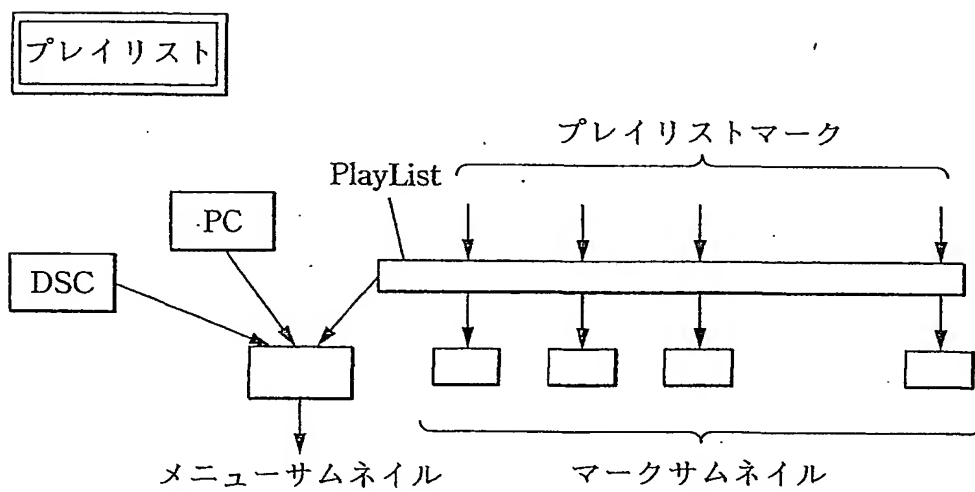


FIG.11

11/118

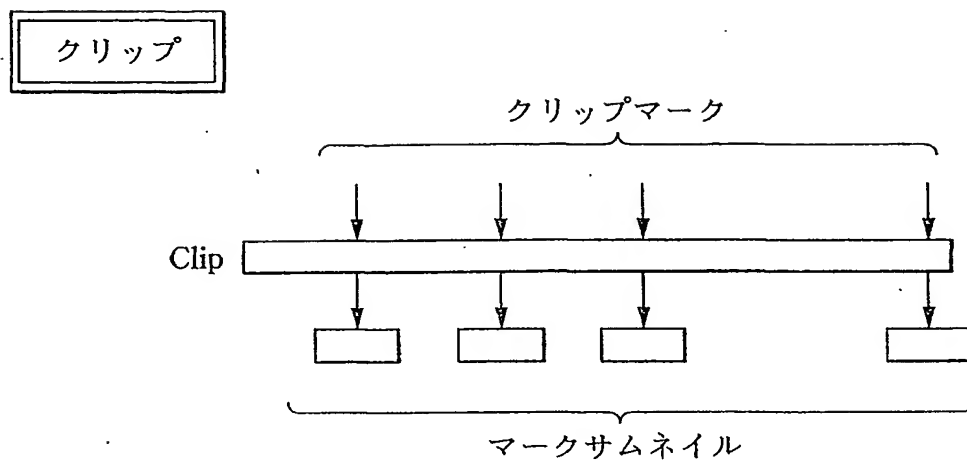


FIG.12

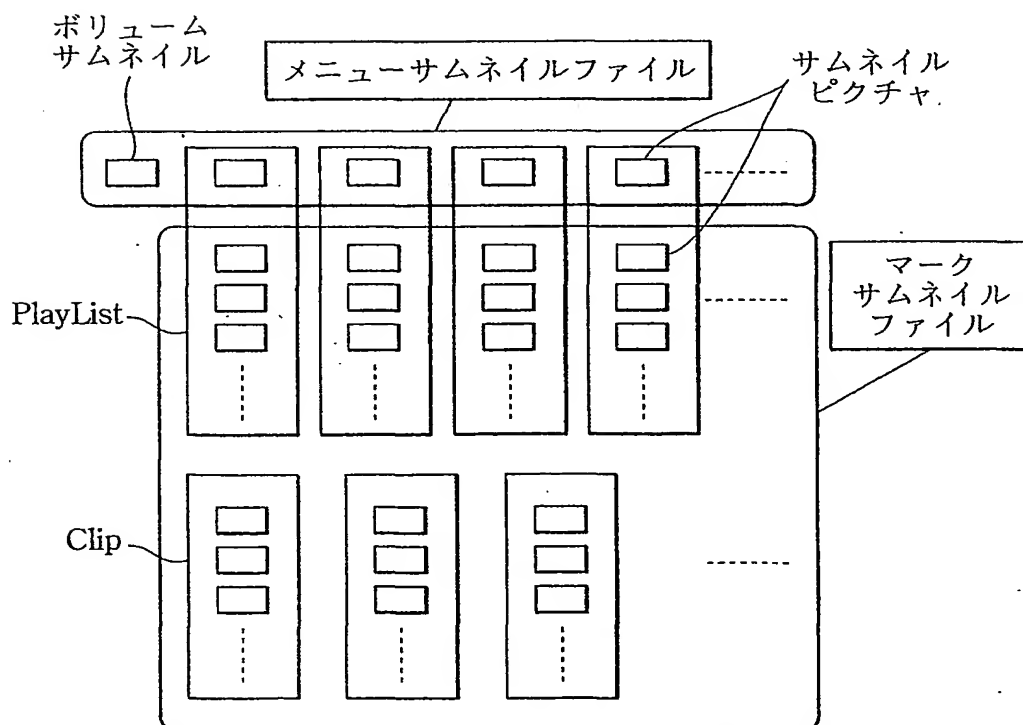


FIG.13

12/118

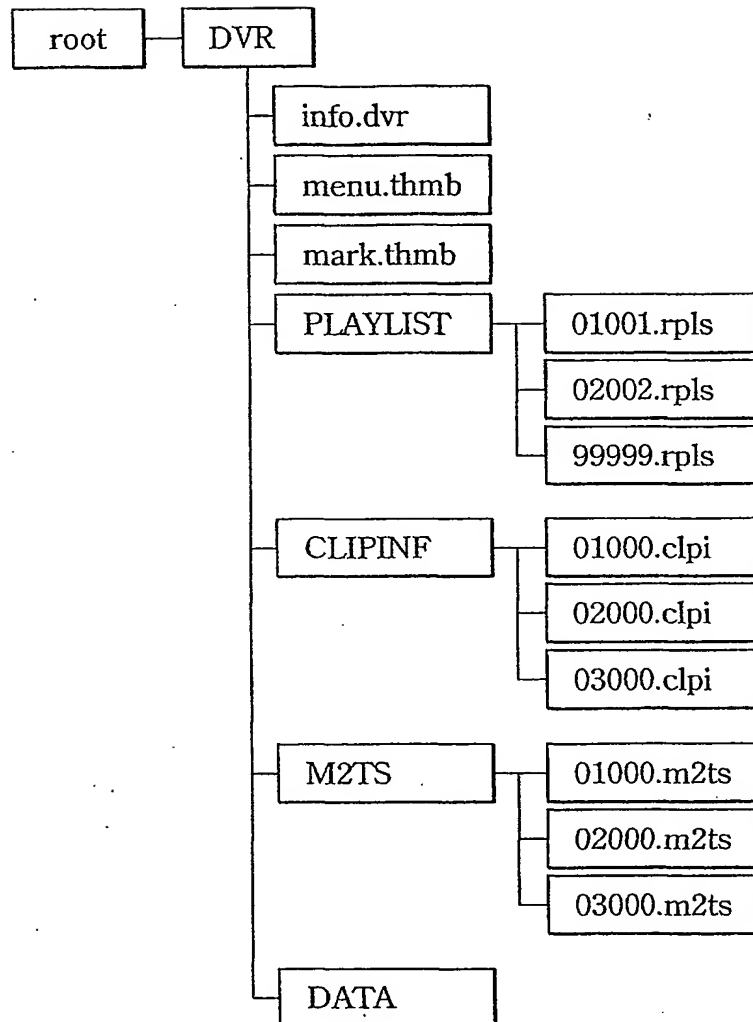


FIG.14

13/118

シンタクス	バイト数	略号
info.dvr {		
TableOfPlayLists_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
DVRVolume()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
TableOfPlayLists()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

FIG.15

14/118

シンタクス	バイト数	略号
DVRVolume(){		
version_number	8*4	bslbf
length	32	uimsbf
ResumeVolume()		
UIAppInfoVolume()		
}		

FIG.16

シンタクス	バイト数	略号
ResumeVolume(){		
reserved	15	bslbf
valid_flag	1	bslbf
resume_PlayList_name	8*10	bslbf
}		

FIG.17

16/118

シンタクス	バイト数	略号
UIAppInfoVolume(){		
character_set	8	bslbf
name_length	8	uimsbf
Volume_name	8*256	bslbf
reserved	15	bslbf
Volume_protect_flag	1	bslbf
PIN	8*4	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

FIG.18

17/118

値	キャラクタ文字符号化
0x00	Reserved
0x01	ISO/IEC 646 (ASCII)
0x02	ISO/IEC 10646-1 (Unicode)
0x03-0xff	Reserved

FIG.19

18/118

シンタクス	バイト数	略号
TableOfPlayLists(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<number_of_PlayLists; i++){		
PlayList_file_name	8*10	bslbf
}		
}		

FIG.20

19/118

シンタクス	バイト数	略号
TableOfPlayLists(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<number_of_PlayLists; i++){		
Playlist_file_name	8*10	bslbf
UIAppInfoPlaylist()		
}		
}		

FIG.21

20/118

シンタクス	バイト数	略号
MakersPrivateData(){		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
mpd_blocks_start_address	32	uimsbf
number_of_maker_entries	16	uimsbf
mpd_block_size	16	uimsbf
number_of_mpd_blocks	16	uimsbf
reserved	16	bslbf
for (i=0; i<number_of_maker_entries; i++){		
maker_ID	16	uimsbf
maker_model_code	16	uimsbf
start_mpd_block_number	16	uimsbf
reserved	16	bslbf
mpd_length	32	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for(j=0; j<number_of_mpd_blocks; j++){		
mpd_block	mpd_block_size*1024*8	
}		
}		
}		

FIG.22

21/118

シンタクス	バイト数	略号
xxxxx.rpls / yyyyy.vpls {		
PlayListMark_Start_address	32	uimbsf
MakersPrivateData_Start_address	32	uimbsf
reserved	192	bslbf
PlayList()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
PlayListMark()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

FIG.23

22/118

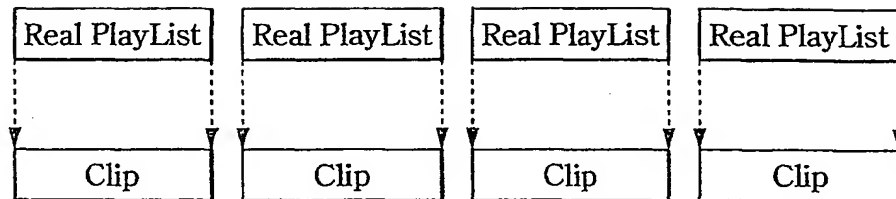


FIG.24A

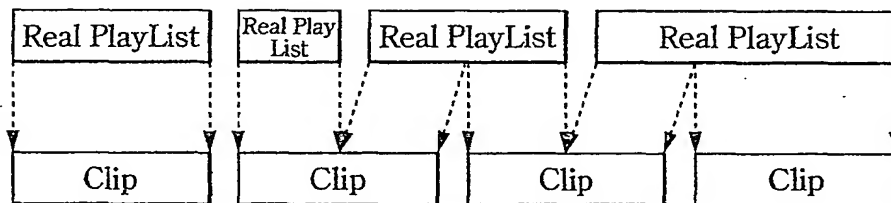


FIG.24B

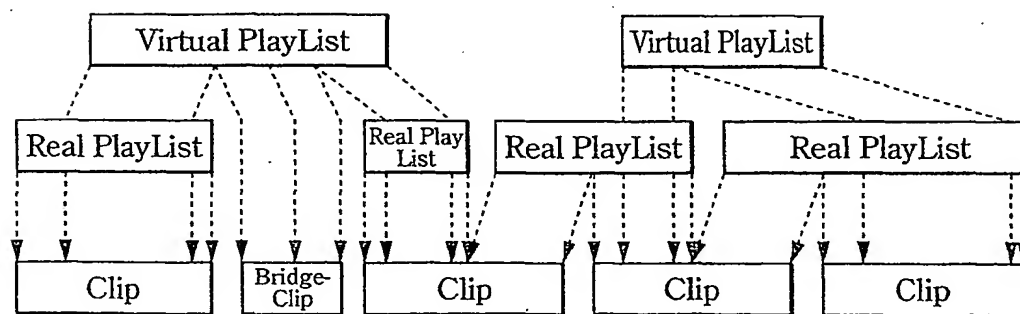


FIG.24C

23/118

シンタクス	バイト数	略号
PlayList{		
version_number	8*4	bslbf
length	32	uimsbf
PlayList_type	8	uimsbf
CPI_type	1	bslbf
reserved	7	bslbf
UIAppInfoPlayList()		
number_of_PlayItems // main path	16	uimsbf
if (<Virtual PlayList>){		
number_of_SubPlayItems // sub path	16	uimsbf
}else{		
reserved	16	bslbf
}		
for (PlayItem_id=0;		
PlayItem_id<number_of_PlayItems;		
PlayItem_id++){		
PlayItem() //main path		
}		
if (<Virtual PlayList>){		
if (CPI_type==0 && PlayList_type==0){		
for (i=0; i<number_of_SubPlayItems; i++)		
SubPlayItem() //sub path		
}		
}		
}		

FIG.25

24/118

PlayList_type	意味
0	AV記録のためのPlayList このPlayListに参照されるすべてのClipは、一つ以上のビデオストリームを含まなければならない。
1	オーディオ記録のためのPlayList このPlayListに参照されるすべてのClipは、一つ以上のオーディオストリームを含まなければならない、そしてビデオストリームを含んではならない。
2-255	reserved

FIG.26

25/118

シンタクス	バイト数	略号
UIAppInfoPlayList20{		
character_set	8	bslbf
name_length	8	uimsbf
PlayList_name	8*256	bslbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
valid_period	4*8	bslbf
maker_id	16	uimsbf
maker_code	16	uimsbf
reserved	11	bslbf
playback_control_flag	1	bslbf
write_protect_flag	1	bslbf
is_played_flag	1	bslbf
archive	2	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

FIG.27

26/118

write_protect_flag	意味
0b	そのPlayListを自由に消去しても良い。
1b	write_protect_flagを除いてそのPlayListの内容は、消去および変更されるべきではない。

FIG.28A

is_played_flag	意味
0b	そのPlayListは、記録されてから一度も再生されたことがない。
1b	PlayListは、記録されてから一度は再生された。

FIG.28B

archive	意味
00b	何も情報が定義されていない。
01b	オリジナル
10b	コピー
11b	reserved

FIG.28C

27/118

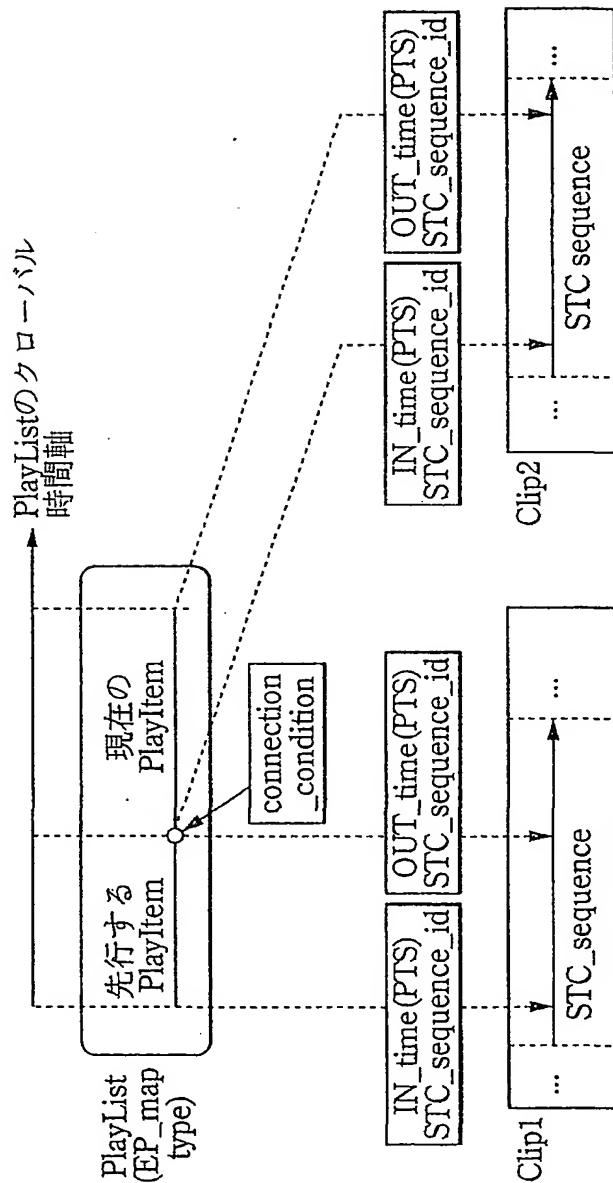


FIG.29

28/118

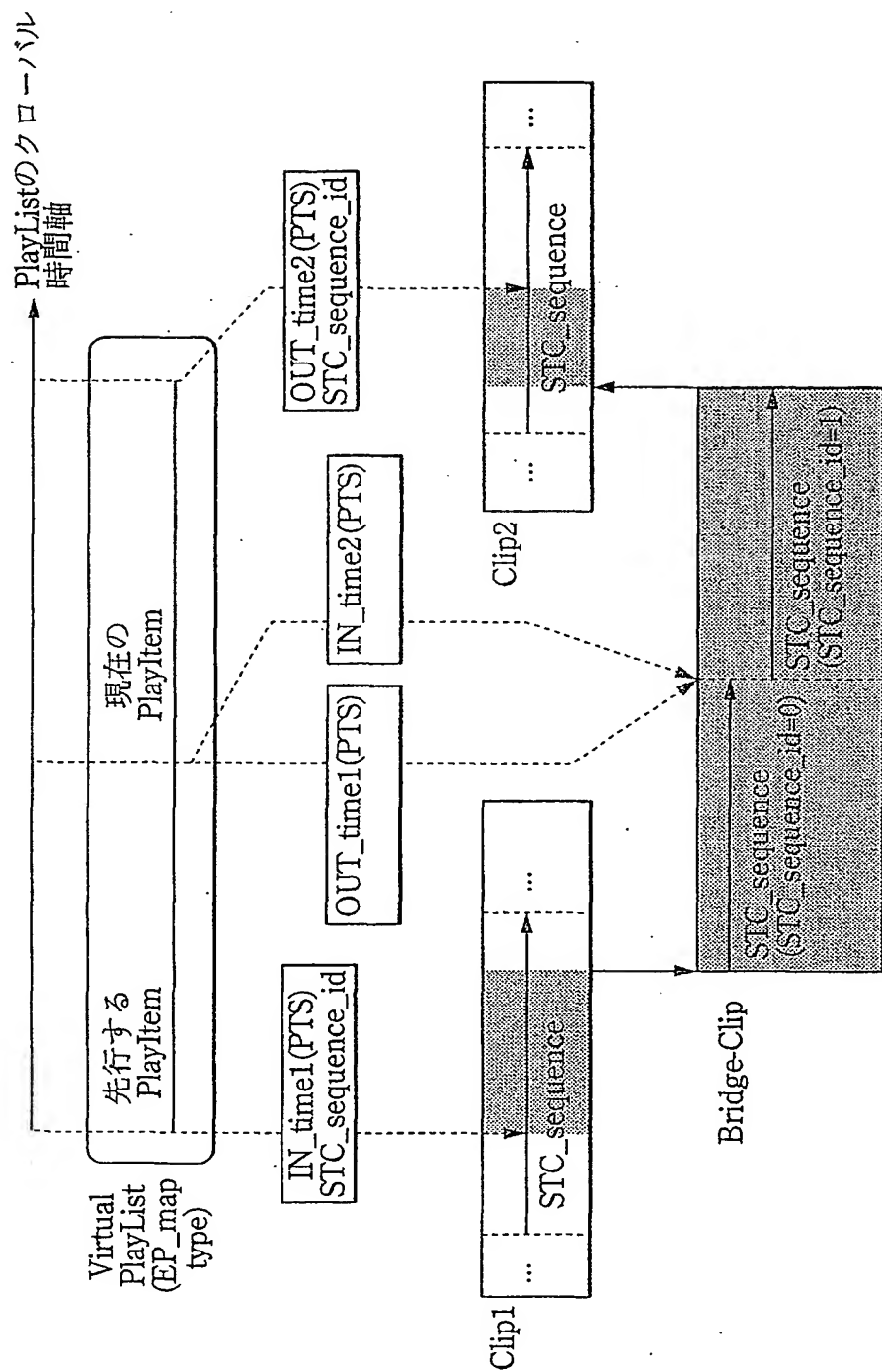


FIG.30

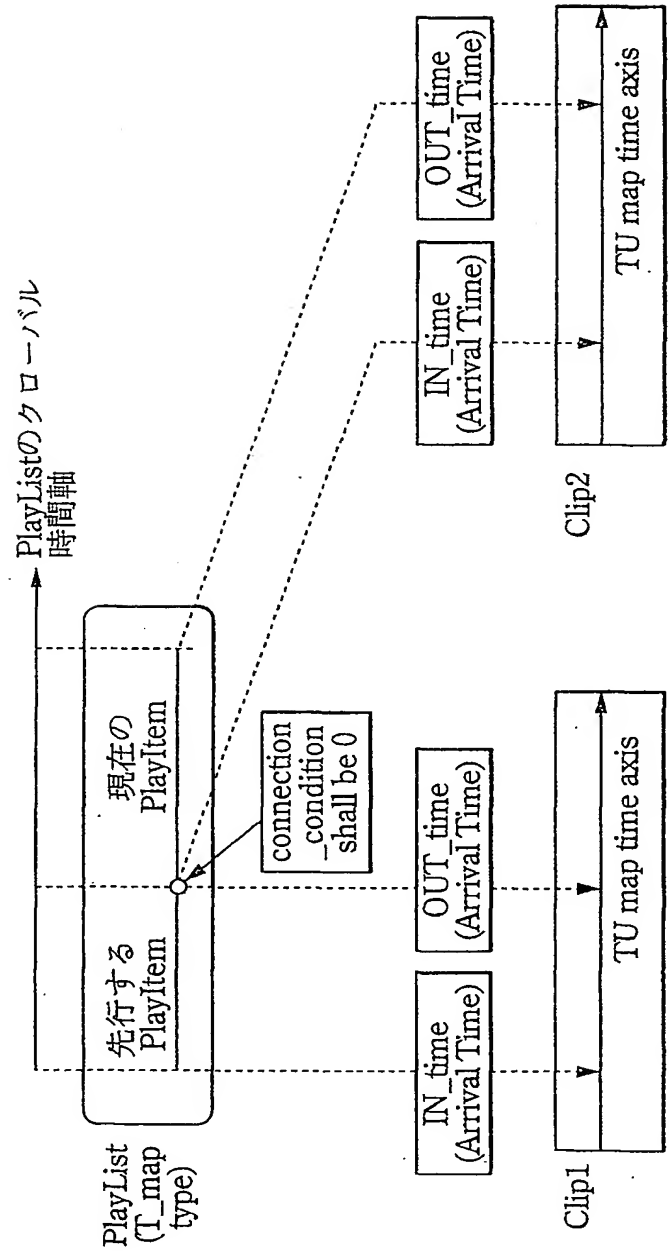


FIG.31

30/118

シンタクス	バイト数	略号
PlayItem(){		
Clip_information_file_name	8*10	bslbf
reserved	24	bslbf
STC_sequence_id	8	uimsbf
IN_time	32	uimsbf
OUT_time	32	uimsbf
reserved	14	bslbf
connection_condition	2	bslbf
if (<Virtual PlayList>){		
if (connection_condition=='10'){		
BridgeSequenceInfo()		
}		
}		
}		

FIG.32

31/118

CPI_type in the PlayList()	IN_timeのセマンティクス
EP_map type	IN_timeは、PlayItemの中で最初のプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示さなければならない。
TU_map type	IN_timeは、TU_map_time_axis上の時刻でなければならない。かつ、IN_timeは、time_unitの精度に丸めて表さなければならない。IN_timeは、次に示す等式により計算される。 $IN_time = TU_start_time \% 2^{32}$

FIG.33

32/118

CPI_type in the PlayList()	OUT_timeのセマンティクス
EP_map type	<p>OUT_timeは、次に示す等式により計算される Presentation_end_TSの値の上位32ビットを示さなければならない。</p> $\text{Presentation_end_TS} = \text{PTS_out} + \text{AU_duration}$ <p>ここで、</p> <p>PTS_outは、PlayItemの中で最後のプレゼンテーションユニットに対応する33ビット長のPTSである。</p> <p>AU_durationは、最後のプレゼンテーションユニットの90kHz単位の表示期間である。</p>
TU_map type	<p>OUT_timeは、TU_map_time_axis上の時刻でなければならない。かつ、OUT_timeは、time_unitの精度に丸めて表さなければならない。OUT_timeは、次に示す等式により計算される。</p> $\text{OUT_time} = \text{TU_start_time} \% 2^{32}$

FIG.34

33/118

connection _condition	意味
00	<ul style="list-style-type: none">・ 先行するPlayItemと現在のPlayItemの接続は、シームレス再生の保証がなされていない。・ PlayListのCPI_typeがTU_map typeである場合、connection_conditionは、この値をセットされねばならない。
01	<ul style="list-style-type: none">・ この状態は、PlayListのCPI_typeがEP_map typeである場合にだけ許される。・ 先行するPlayItemと現在のPlayItemは、システムタイムベース（STCベース）の不連続点があるために分割されていることを表す。
10	<ul style="list-style-type: none">・ この状態は、PlayListのCPI_typeがEP_map typeである場合にだけ許される。・ この状態は、Virtual PlayListに対してだけ許される。・ 先行するPlayItemと現在のPlayItemとの接続は、シームレス再生の保証がなされている。・ 先行するPlayItemと現在のPlayItemは、BridgeSequenceを使用して接続されており、DVR MPEG-2トランスポートストリームは、後述するDVR-STDに従っていなければならない。
11	<ul style="list-style-type: none">・ この状態は、PlayListのCPI_typeがEP_map typeである場合にだけ許される。・ 先行するPlayItemと現在のPlayItemは、シームレス再生の保証がなされている。・ 先行するPlayItemと現在のPlayItemは、BridgeSequenceを使用しないで接続されており、DVR MPEG-2トランスポートストリームは、後述するDVR-STDに従っていなければならない。

FIG.35

34/118

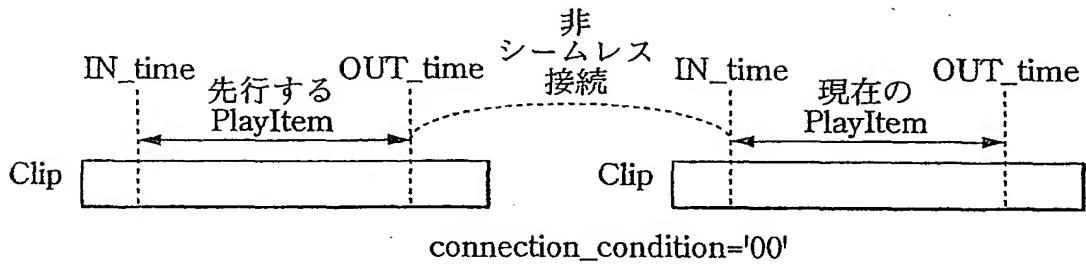


FIG.36A

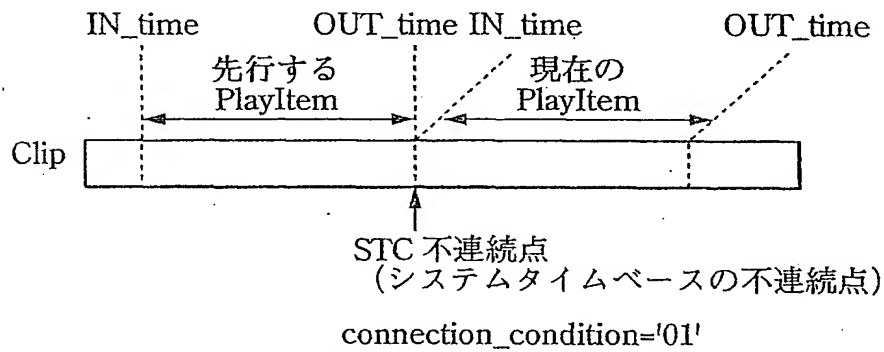


FIG.36B

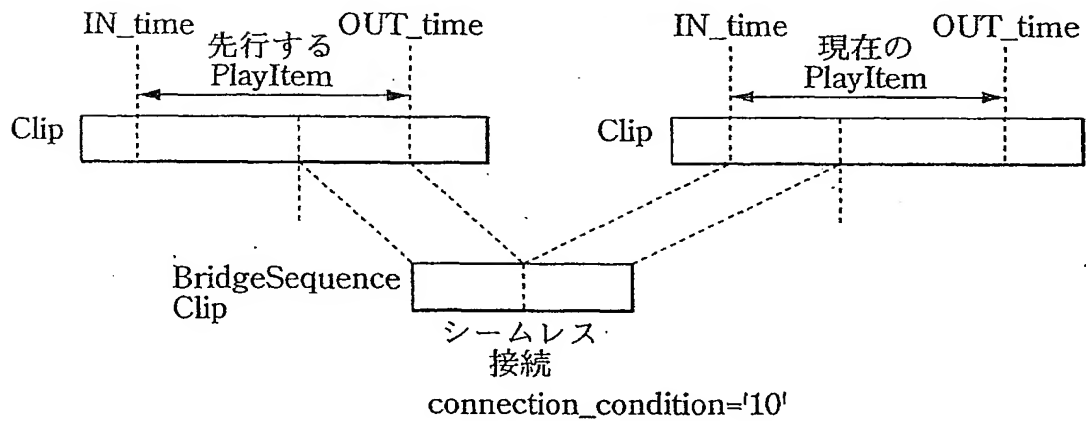


FIG.36C

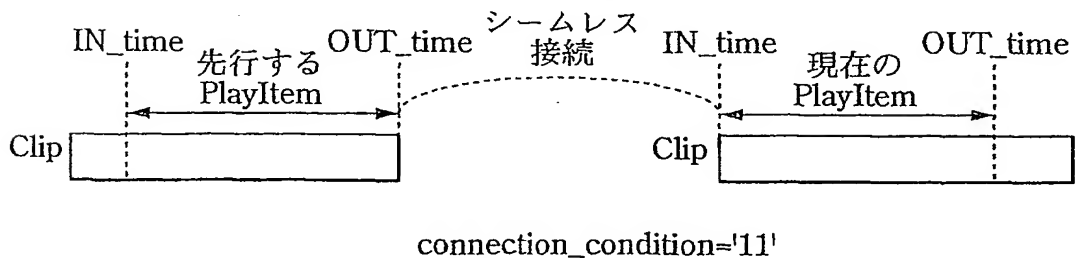


FIG.36D

35/118

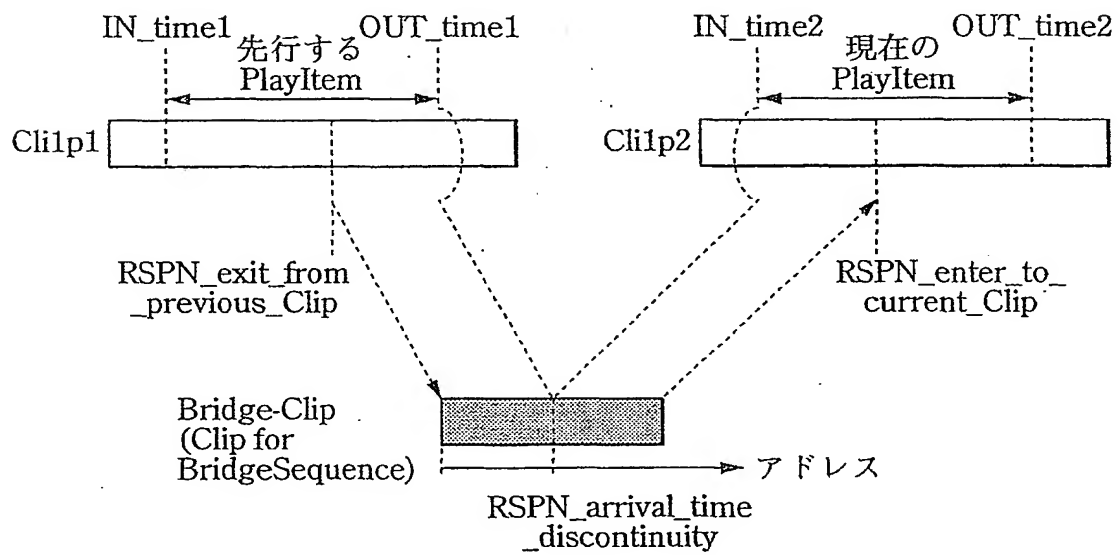


FIG.37

36/118

シンタクス	バイト数	略号
BridgeSequenceInfo() {		
Bridge_Clip_information_file_name	8*10	bslbf
RSPN_exit_from_previous_Clip	32	uimsbf
RSPN_enter_to_current_Clip	32	uimsbf
}		

FIG.38

37/118

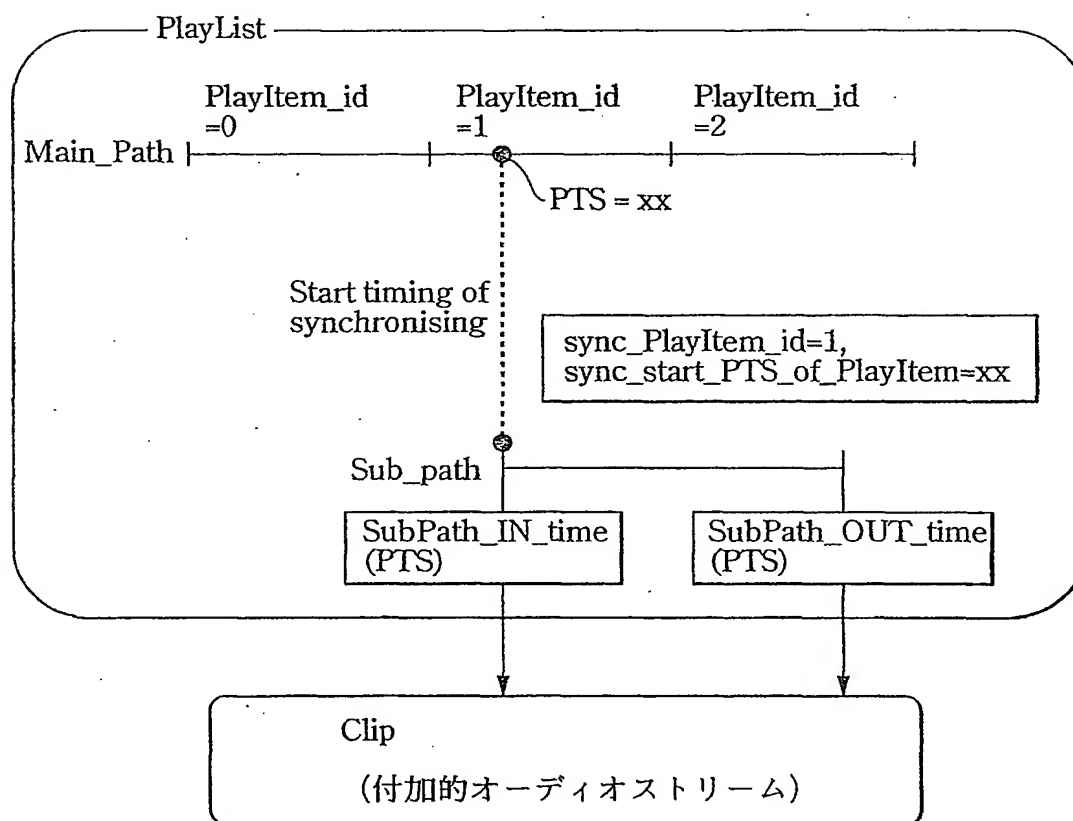


FIG.39

38/118

シンタクス	バイト数	略号
SubPlayItem(){		
Clip_Information_file_name	8*10	bslbf
SubPath_type	8	bslbf
sync_PlayItem_id	8	uimsbf
sync_start_PTS_of_PlayItem	32	uimsbf
SubPath_IN_time	32	uimsbf
SubPath_OUT_time	32	uimsbf
}		

FIG.40

39/118

SubPath_type	意味
0x00	Auxiliary audio steam path
0x01-0xff	reserved

FIG.41

40/118

シンタクス	バイト数	略号
PlayListMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for (i=0;i<number_of_PlayList_marks;i++){		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
PlayItem_id	8	uimsbf
reserved	24	uimsbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

FIG.42

41/118

Mark_type	意味	コメント
0x00	resume-mark	再生リジュームポイント。PlayListMark()において定義される再生リジュームポイントの数は、0または1でなければならない。
0x01	book-mark	PlayListの再生エントリーポイント。このマークは、ユーザがセットすることができ、例えば、お気に入りのシーンの開始点を指定するマークに使う。
0x02	skip-mark	スキップマークポイント。このポイントからプログラムの最後まで、プレーヤはプログラムをスキップする。PlayListMark()において定義されるスキップマークポイントの数は、0または1でなければならない。
0x03-0x8F	reserved	
0x90-0xFF	reserved	Reserved for ClipMark()

FIG.43

42/118

CPI_type in the PlayList()	mark_time_stampのセマンティクス
EP_map type	mark_time_stampは、マークで参照されるプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示さなければならない。
TU_map type	mark_time_stampは、TU_map_time_axis上の時刻でなければならない。かつ、mark_time_stampは、time_unitの精度に丸めて表さなければならない。mark_time_stampは、次に示す等式により計算される。 $\text{mark_time_stamp} = \text{TU_start_time} \% 2^{32}$

FIG.44

43/118

シンタクス	バイト数	略号
zzzzz.clpi {		
STC_Info_Start_address	32	uimbsf
ProgramInfo_Start_address	32	uimbsf
CPI_Start_address	32	uimbsf
ClipMark_Start_address	32	uimbsf
MakersPrivateData_Start_address	32	uimbsf
reserved	96	bslbf
ClipInfo()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
STC_Info()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
ProgramInfo()		
for (i=0;i<N3;i++){		
padding_word	16	bslbf
}		
CPI()		
for (i=0;i<N4;i++){		
padding_word	16	bslbf
}		
ClipMark()		
for (i=0;i<N5;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

FIG.45

44/118

シンタクス	バイト数	略号
ClipInfo(){		
version_number	8*4	bslbf
length	32	uimsbf
Clip_stream_type	8	bslbf
offset_SPN	32	uimsbf
TS_recording_rate	24	uimsbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
reserved	7	bslbf
time_controlled_flag	1	bslbf
TS_average_rate	24	uimsbf
if (Clip_stream_type==1) // Bridge-Clip AV stream		
RSPN_arrival_time_discontinuity	32	uimsbf
else		
reserved	32	bslbf
reserved_for_system_use	144	bslbf
reserved	11	bslbf
is_format_identifier_valid	1	bslbf
is_original_network_ID_valid	1	bslbf
is_transport_stream_ID_valid	1	bslbf
is_service_ID_valid	1	bslbf
is_country_code_valid	1	bslbf
format_identifier	32	bslbf
original_network_ID	16	uimsbf
transport_stream_ID	16	uimsbf
service_ID	16	uimsbf
country_code	24	bslbf
stream_format_name	16*8	bslbf
reserved_for_fortune_use	256	bslbf
}		

FIG.46

45/118

Clip_stream_type	意味
0	Clip AV ストリーム
1	Bridge-Clip AV ストリーム
2-255	Reserved

FIG.47

46/118

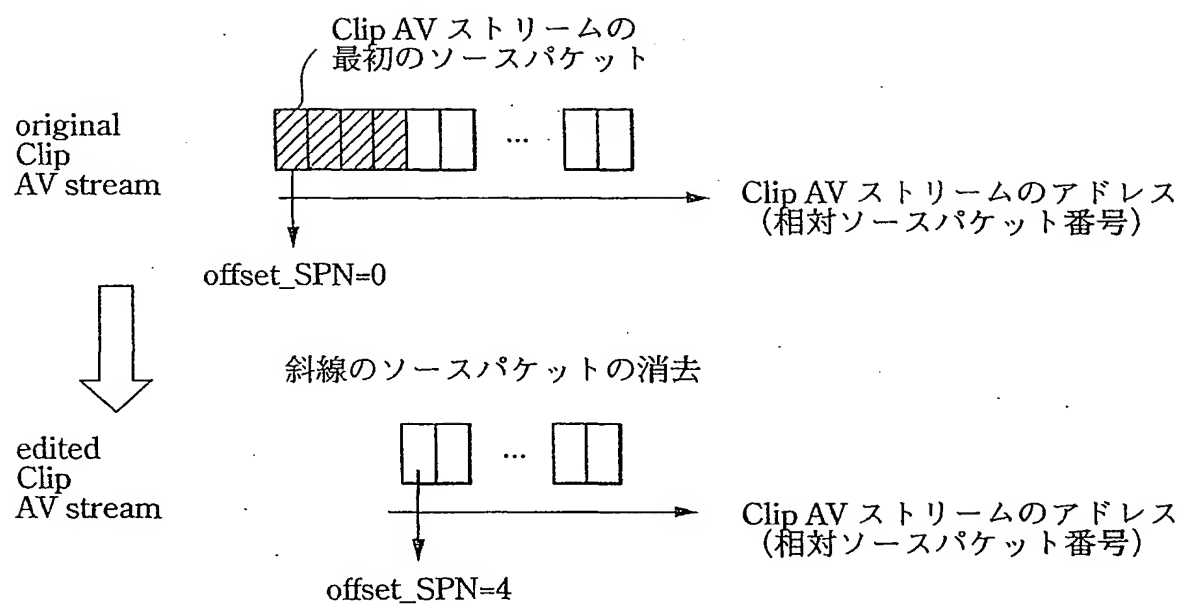


FIG.48

47/118

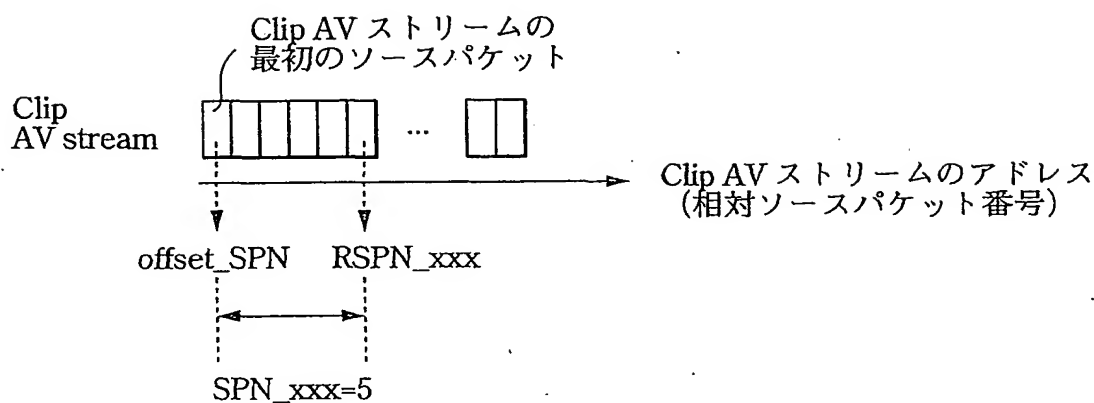


FIG.49

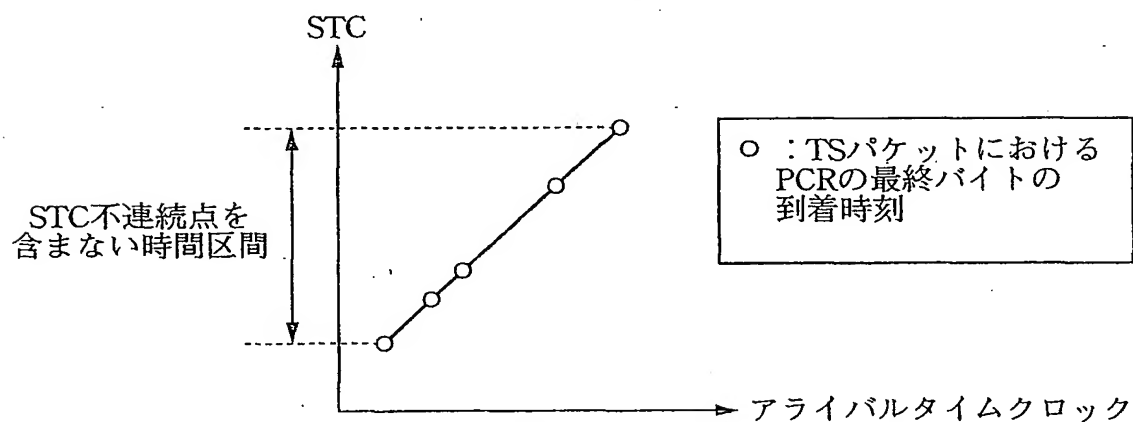


FIG.50A

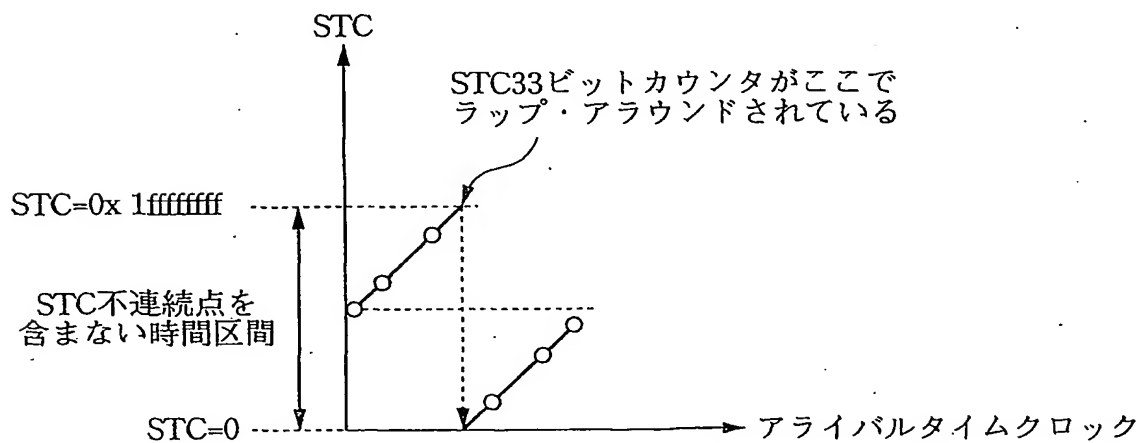


FIG.50B

48/118

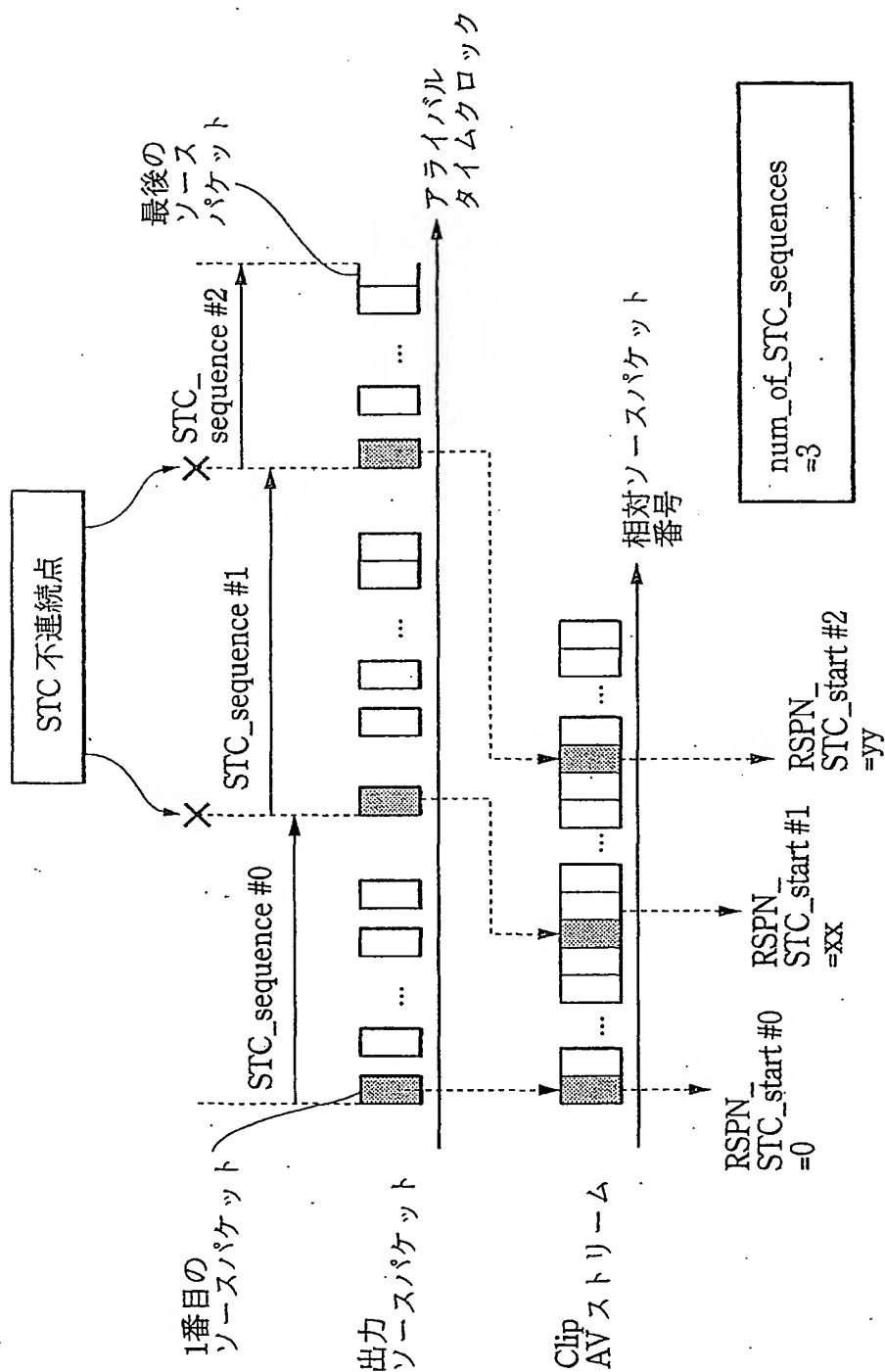


FIG.51

49/118

シンタクス	バイト数	略号
STC_Info(){		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
reserved	8	bslbf
num_of_STC_sequences	8	uimsbf
for (STC_sequence_id=0; STC_sequence_id<num_of_STC_sequences; STC_sequence_id++){		
resereved	32	bslbf
RSPN_STC_start	32	uimsbf
}		
}		
}		

FIG.52

50/118

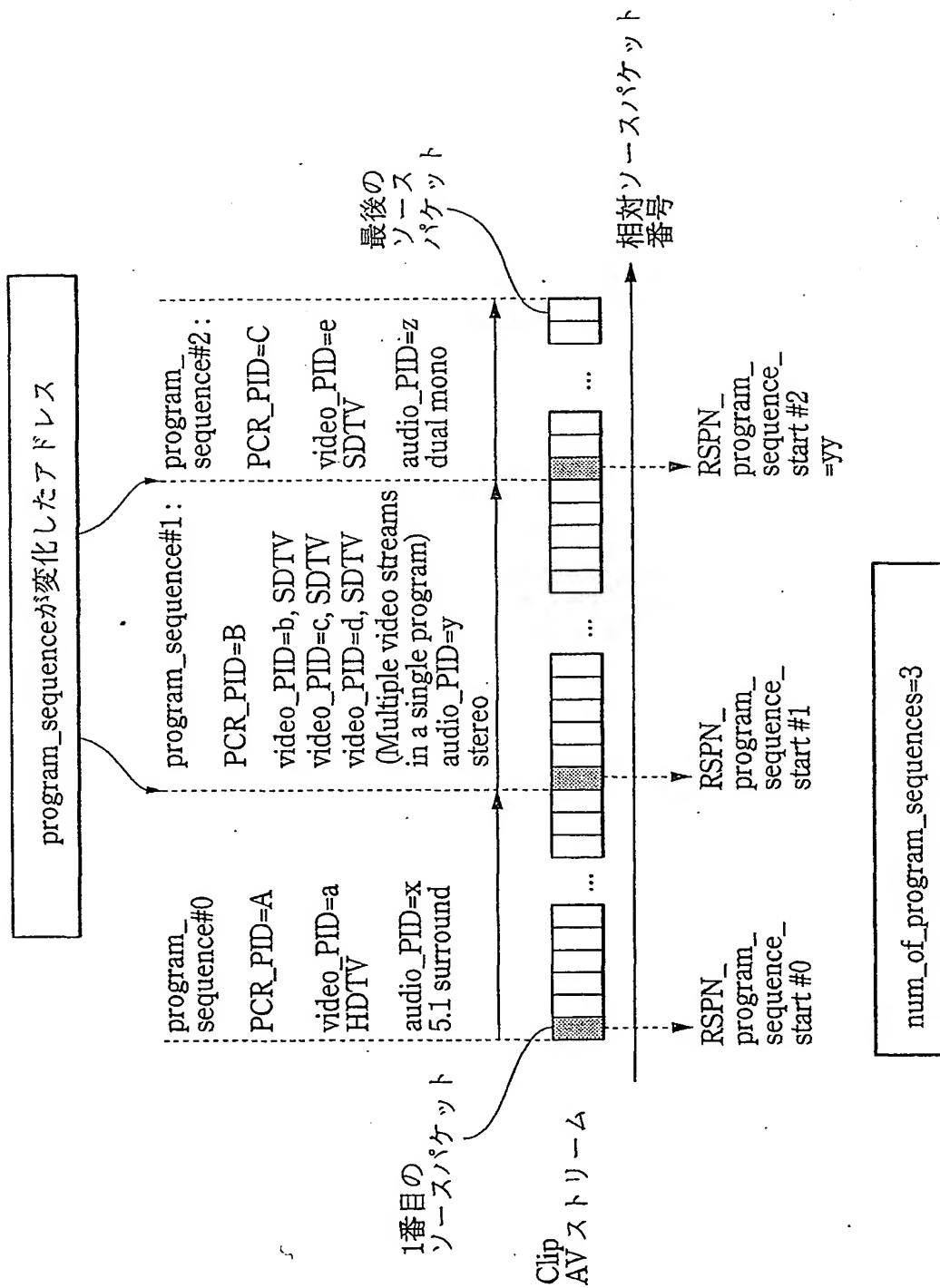


FIG.53

51/118

シンタクス	バイト数	略号
ProgramInfo(){		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
reserved	8	bslbf
number_of_program_sequences	8	uimsbf
for (i=0;i<number_of_program_sequences;i++){		
RSPN_program_sequence_start	32	uimsbf
reserved	48	bslbf
PCR_PID	16	bslbf
number_of_videos	8	uimsbf
number_of_audios	8	uimsbf
for (k=0;k<number_of_videos;k++){		
video_stream_PID	16	bslbf
VideoCodingInfo()		
}		
for (k=0;k<number_of_audios;k++){		
audio_stream_PID	16	bslbf
AudioCodingInfo()		
}		
}		
}		
}		

FIG.54

52/118

シンタクス	バイト数	略号
VideoCodingInfo() {		
video_format	8	uimsbf
frame_rate	8	uimsbf
display_aspect_ratio	8	uimsbf
reserved	8	bslbf
}		

FIG.55

53/118

video_format	意味
0	480i
1	576i
2	480p(including 640×480p format)
3	1080i
4	720p
5	1080p
6-254	reserved
255	No information

FIG.56

frame_rate	意味
0	forbidden
1	24 000/1001 (23.976...)
2	24
3	25
4	30 000/1001 (29.97..)
5	30
6	50
7	60 000/1001 (59.94..)
8	60
9-254	reserved
255	No information

FIG.57

54/118

display_aspect_ratio	意味
0	forbidden
1	reserved
2	4:3 display aspect ratio
3	16:9 display aspect ration
4-254	reserved
255	No information

FIG.58

55/118

シンタクス	バイト数	略号
AudioCodingInfo() {		
audio_format	8	uimbsf
audio_component_type	8	uimbsf
sampling_frequency	8	uimbsf
reserved	8	bslbf
}		

FIG.59

56/118

audio_coding	意味
0	MPEG-1 audio layer I or II
1	Dolby AC-3 audio
2	MPEG-2 AAC
3	MPEG-2 multi-channel audio, backward compatible to MPEG-1
4	SESF LPCM audio
5-254	reserved
255	No information

FIG.60

57/118

audio_component_type	意味
0	single mono channel
1	dual mono channel
2	stereo (2-channel)
3	multi-lingual, multi-channel
4	surround sound
5	audio description for the visually impaired
6	audio for the hard of hearing
7-254	reserved
255	No information

FIG.61

sampling_frequency	意味
0	48 kHz
1	44.1 kHz
2	32 kHz
3-254	reserved
255	No information

FIG.62

58/118

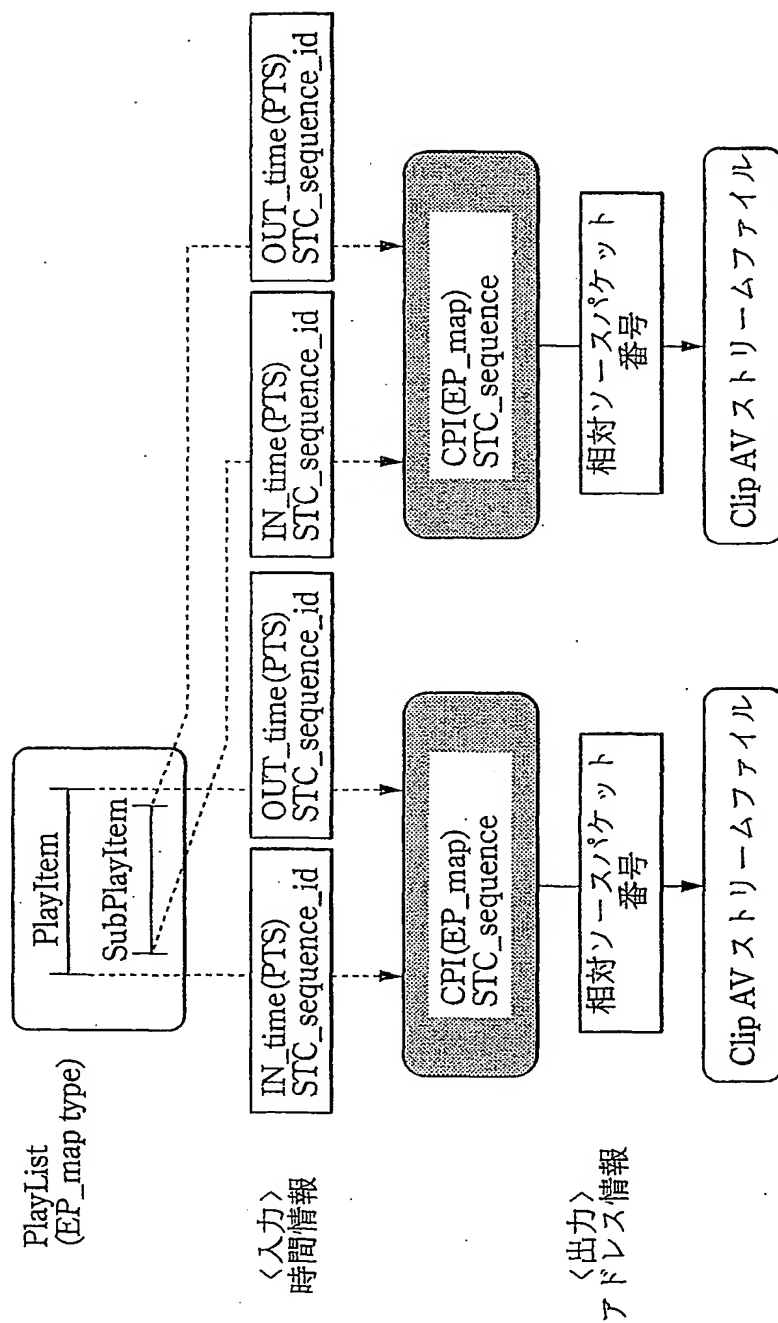


FIG. 63

59/118

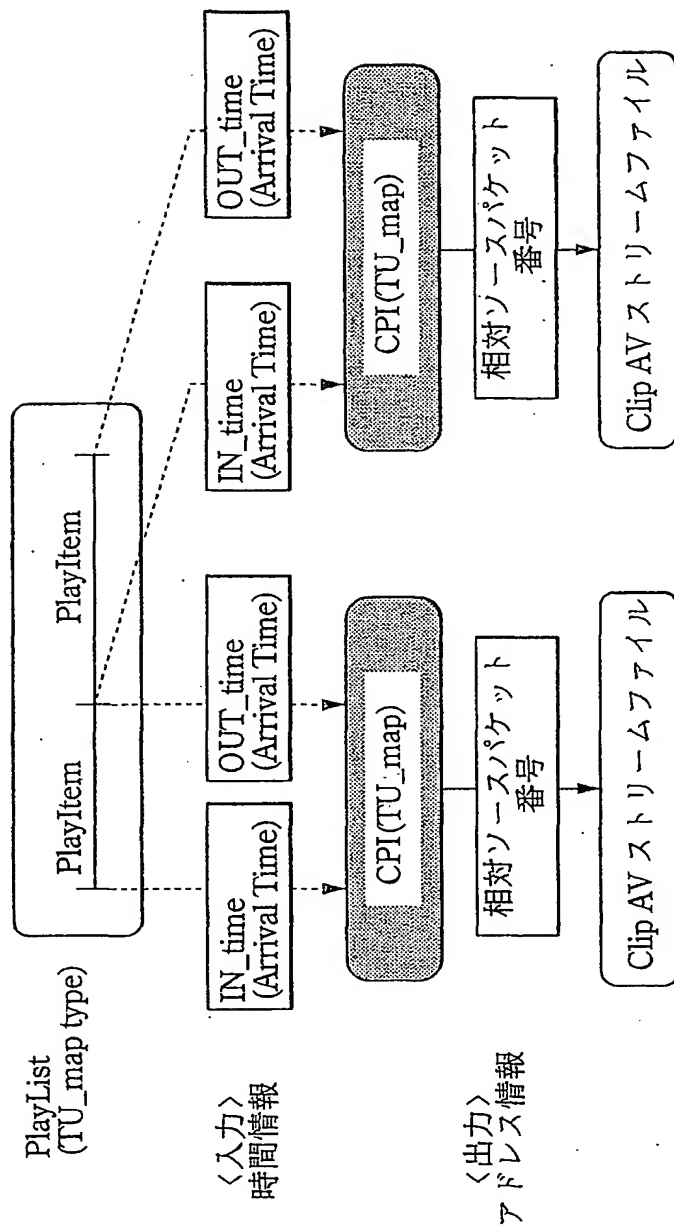


FIG. 64

60/118

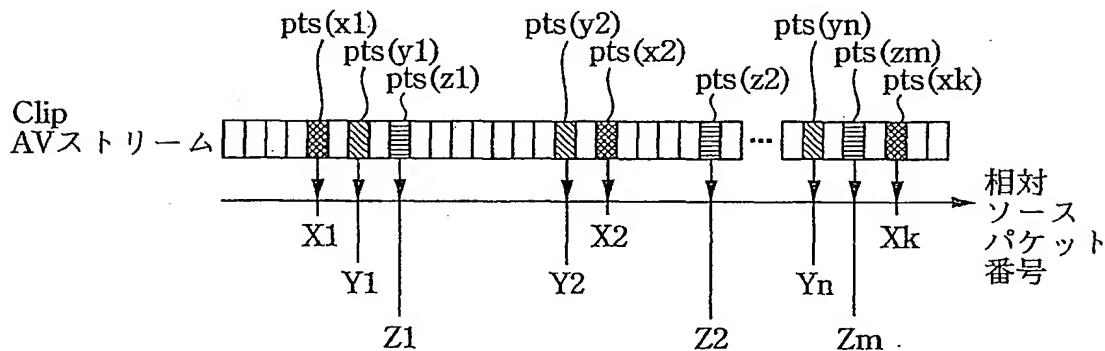
シンタクス	バイト数	略号
CPI0{		
version_number	8*4	bslbf
length	32	uimsbf
reserved	15	bslbf
CPI_type	1	bslbf
if (CPI_type==0)		
EP_map()		
else		
TU_map()		
}		

FIG.65

61/118

CPI_type	意味
0	EP map type
1	TU map type

FIG.66



- ▨ : シーケンスヘッダvideo_PID=xの第1バイト目を含むソースパケット
- ▤ : シーケンスヘッダvideo_PID=yの第1バイト目を含むソースパケット
- ▥ : シーケンスヘッダvideo_PID=zの第1バイト目を含むソースパケット

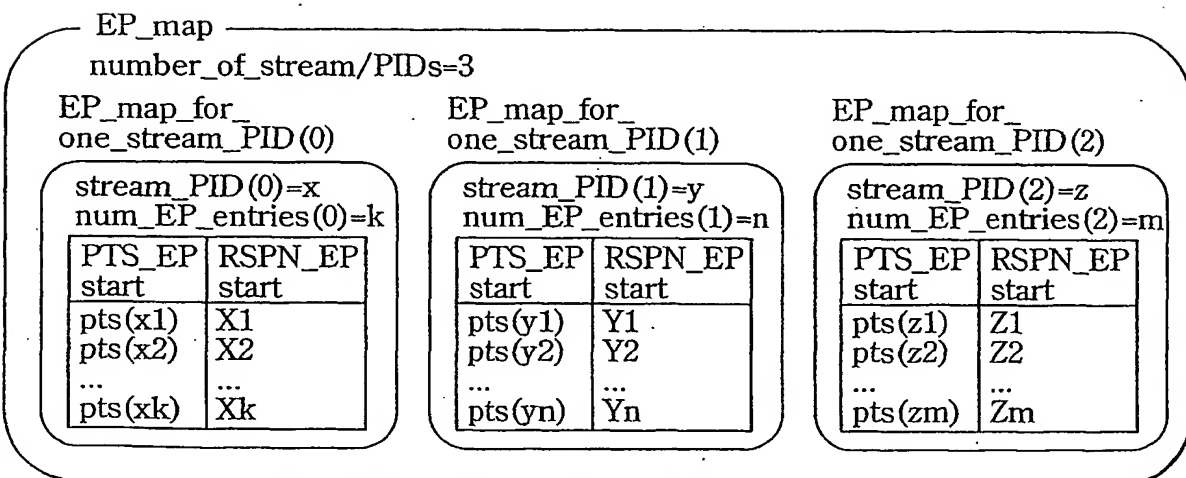
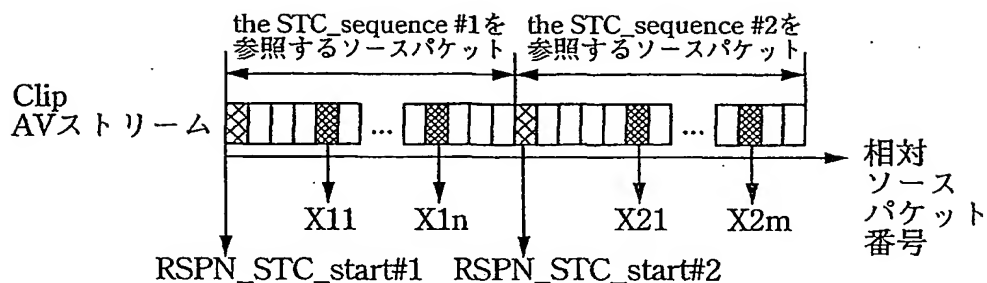


FIG.67

62/118



▨ : シーケンスヘッダ video_PID=x の第 1 バイト目を含むソースパケット

▤ : RSPN_STC_start によって参照されるソースパケット (the STC_into において定義される)

EP_map_for_one_stream_PID
video_PID=x

PTS_EP start	RSPN_EP start	
pts(x11)	X11) STC_sequence #1に属するデータ
...	...	
pts(x1n)	X1n) STC_sequence #2に属するデータ
...	...	
pts(x21)	X21) STC_sequence #2に属するデータ
pts(x2m)	X2m	

boundary

RSPN_STC_start #2 < X21

FIG.68

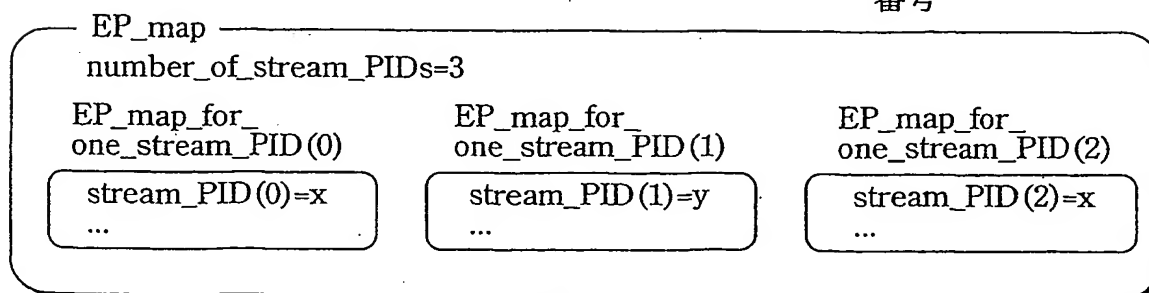
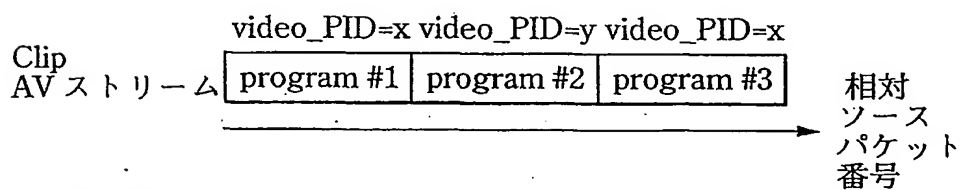


FIG.69

63/118

シンタクス	バイト数	略号
EP_map0{		
reserved	12	bslbf
EP_type	4	uimsbf
number_of_stream_PIDs	16	uimsbf
for (k=0;k<number_of_stream_PIDs;k++){		
stream_PID(k)	16	bslbf
num_EP_entries(k)	32	uimsbf
EP_map_for_one_stream_PID_Start_address(k)	32	uimsbf
}		
for (i=0;i<X;i++){		
padding_word	16	bslbf
}		
for (k=0;k<number_of_stream_PIDs;k++){		
EP_map_for_one_stream_PID(num_EP_entries(k))		
for (i=0;i<Y;i++){		
padding_word	16	bslbf
}		
}		
}		

FIG.70

64/118

EP_type	意味
0	video
1	audio
2-15	reserved

FIG.71

65/118

シンタクス	バイト数	略号
EP_map_for_one_stream_PID(N) {		
for (i=0;i<N;i++){		
PTS_EP_start	32	uimsbf
RSPN_EP_start	32	uimsbf
}		
}		

FIG.72

66/118

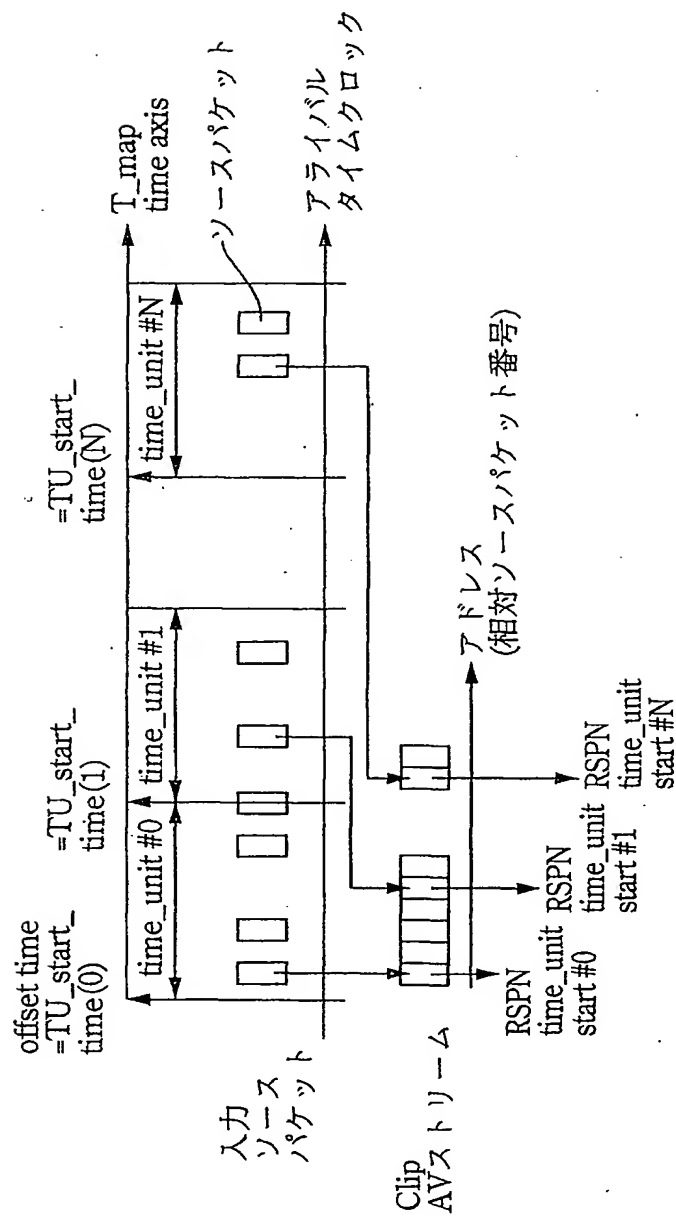


FIG.73

67/118

シンタクス	バイト数	略号
TU_map() {		
offset_time	32	bslbf
time_unit_size	32	uimsbf
number_of_time_unit_entries	32	uimsbf
for (k=0;k<number_of_time_unit_entries;k++)		
RSPN_time_unit_start	32	uimsbf
}		

FIG.74

68/118

シンタクス	バイト数	略号
ClipMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<number_of_clip_marks; i++){		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
STC_sequence_id	8	uimsbf
reserved	24	bslbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

FIG.75

69/118

Mark_type	意味	コメント
0x00-0x8F	reserved	Reserved for PlayListMark()
0x90	Event-start mark	番組の開始ポイントを示すマーク点
0x91	Local event-start mark	番組の中の局所的な場面を示すマーク点
0x92	Scene-start mark	シーンチェンジポイントを示すマーク点
0x93-0xFF	reserved	

FIG.76

70/118

CPI_type in the CPI0	mark_time_stampのセマンティクス
EP_map type	mark_time_stampは、マークで参照されるプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示さなければならない。
TU_map type	<p>mark_time_stampは、TU_map_time_axis上の時刻でなければならない。かつ、mark_time_stampは、time_unitの精度に丸めて表さなければならない。mark_time_stampは、次に示す等式により計算される。</p> $\text{mark_time_stamp} = \text{TU_start_time} \% 2^{32}$

FIG.77

71/118

シンタクス	バイト数	略号
ClipMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<number_of_Clip_marks; i++){		
reserved	8	bslbf
mark_type	8	bslbf
reserved_for_MakerID	16	bslbf
<i>mark_entry()</i>		
<i>representative_picture_entry()</i>		
ref_thumbnail_index	16	uimsbf
}		
}		

FIG.78

Mark_type	意味	コメント
0x00-0x8F	reserved	Reserved for PlayListMark()
0x90	Event-start mark	番組の開始ポイントを示すマーク点
0x91	Local event-start mark	番組の中の局所的な場面を示すマーク点
0x92	Scene-start mark	シーン開始ポイントを示すマーク点
0x93	Scene-end mark	シーン終了ポイントを示すマーク点
0x94	CM-start mark	CM開始ポイントを示すマーク点
0x95	CM-end mark	CM終了ポイントを示すマーク点
0x96-0xBF	DVRフォーマットが、 ClipMarkを将来、拡張 するのために予約さ れている	
0xC0-0xFF	メーカー独自のアプリ ケーションで利用する マークに割り当て可能	

FIG.79

72/118

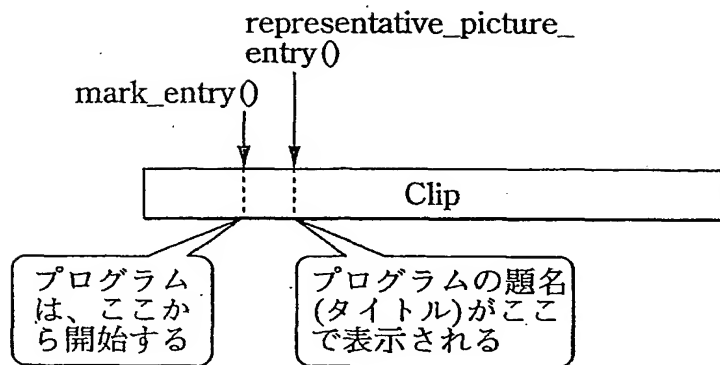


FIG.80

シンタクス	バイト数	略号
mark_entry()/representative_picture_entry(){		
mark_time_stamp	32	uimsbf
STC_sequence_id	8	uimsbf
reserved	24	bslbf
}		

FIG.81

シンタクス	バイト数	略号
mark_entry()/representative_picture_entry(){		
RSPN_ref_EP_start	32	uimsbf
offset_num_pictures	32	uimsbf
}		

FIG.82

73/118

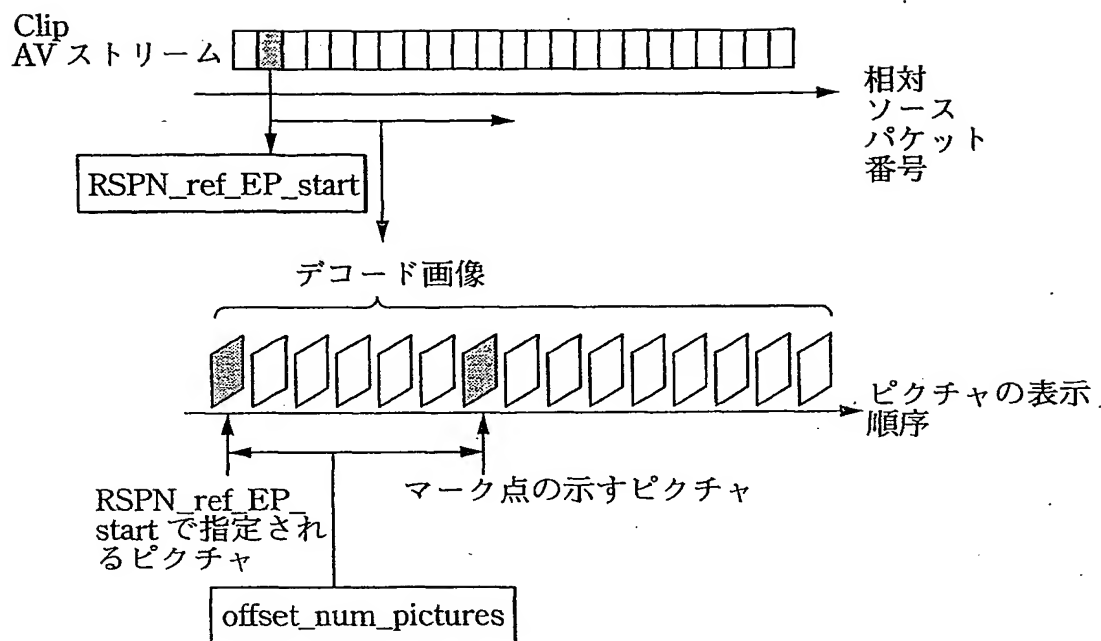
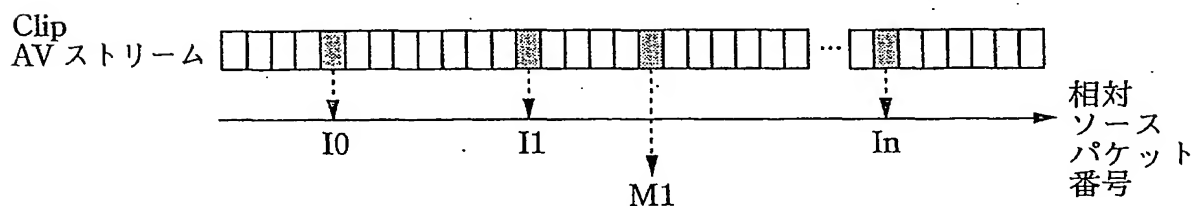


FIG.83

シンタクス	バイト数	略号
mark_entry()/representative_picture_entry(){		
RSPN_mark_point	32	uimsbf
}		

FIG.84



- : EP_map によって指定されるアドレスにあるソースパケット。
このソースパケットから、Iピクチャが開始している。
- : ClipMarkで指定されるアドレスにあるソースパケット。
このソースパケットからマークで指定されるピクチャが開始している。

FIG.85

74/118

シンタクス	バイト数	略号
menu.thmb/mark.thmb() {		
reserved	256	bslbf
Thumbnail()		
for (i=0;i<N1;i++)		
padding_word	16	bslbf
}		

FIG.86

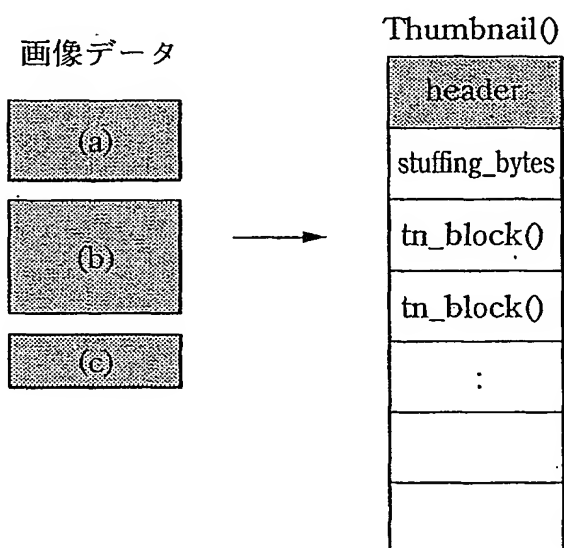
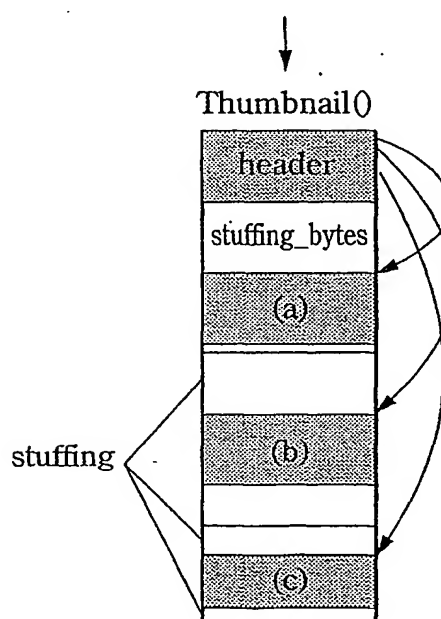
75/118

シンタクス	バイト数	略号
Thumbnail(){		
version_number	8*4	char
length	32	uimsbf
if (length !=0){		
tn_blocks_start_address	32	bslbf
number_of_thumbnails	16	uimsbf
tn_block_size	16	uimsbf
number_of_tn_blocks	16	uimsbf
reserved	16	bslbf
for (i=0; i<number_of_thumbnails; i++){		
thumbnail_index	16	uimsbf
thumbnail_picture_format	8	bslbf
reserved	8	bslbf
picture_data_size	32	uimsbf
start_tn_block_number	16	uimsbf
x_picture_length	16	uimsbf
y_picture_length	16	uimsbf
reserved	16	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for(k=0; k<number_of_tn_blocks; k++){		
tn_block	tn_block_size*1024*8	
}		
}		

FIG.87

76/118

Thumbnail_picture_format	意味
0x00	MPEG-2 Video I-picture
0x01	DCF (restricted JPEG)
0x02	PNG
0x03-0xff	reserved

FIG.88**FIG.89A****FIG.89B**

77/118

DVR MPEG-2 トランスポートストリーム

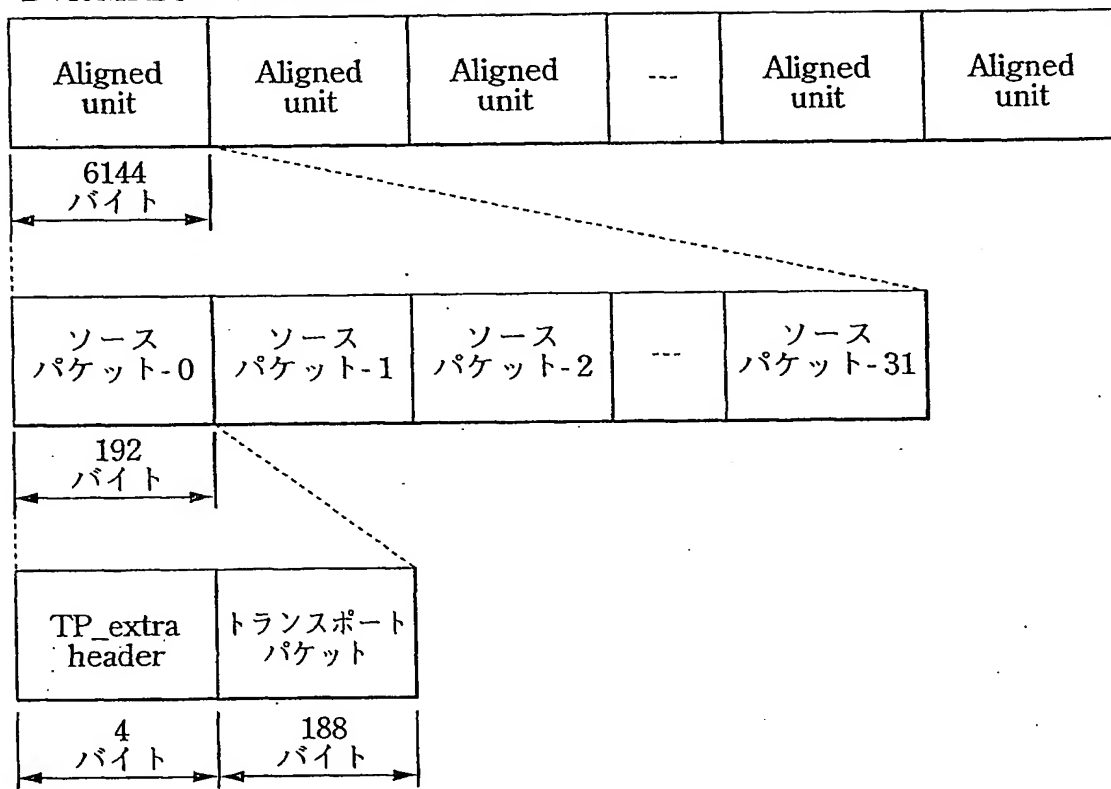


FIG.90

78/118

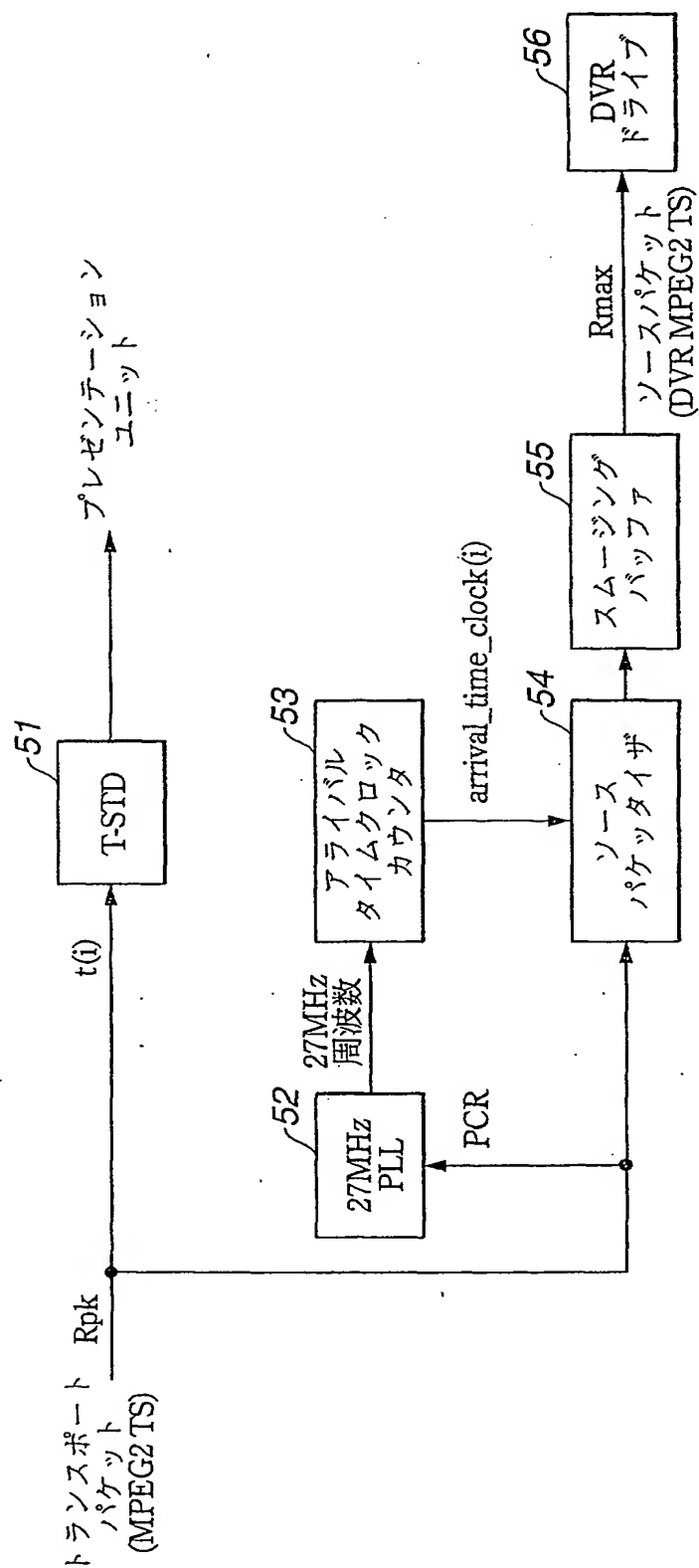


FIG.91

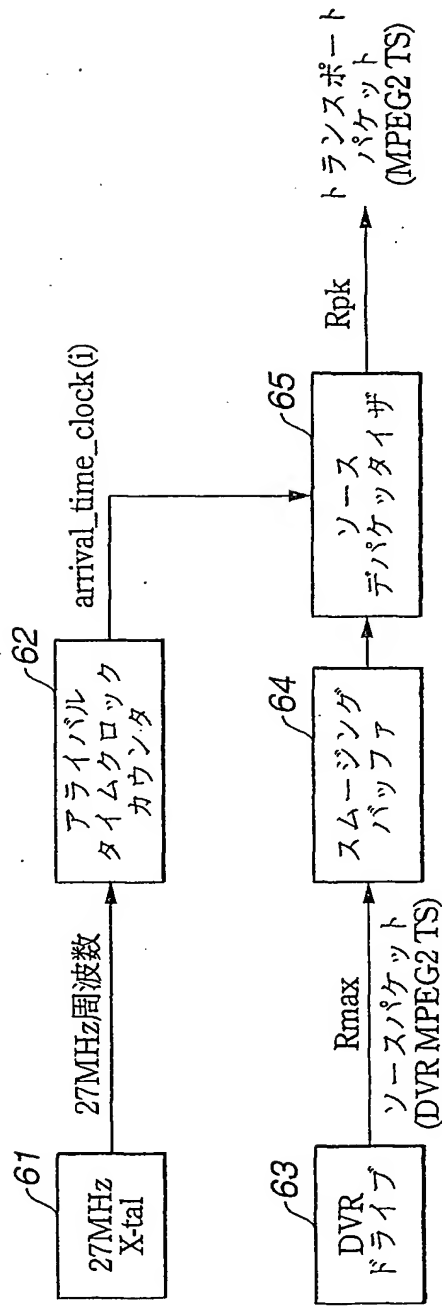


FIG.92

80/118

シンタクス	バイト数	略号
source_packet() {		
TP_extra_header()		
trasport_packet()		
}		

FIG.93

81/118

シンタクス	バイト数	略号
TP_extra_header() {		
copy_permission_indicator	2	uimsbf
arrival_time_stamp	30	uimsbf
}		

FIG.94

82/118

copy_permission _indicator	意味
00	copy free
01	no more copy
10	copy once
11	copy prohibited

FIG.95

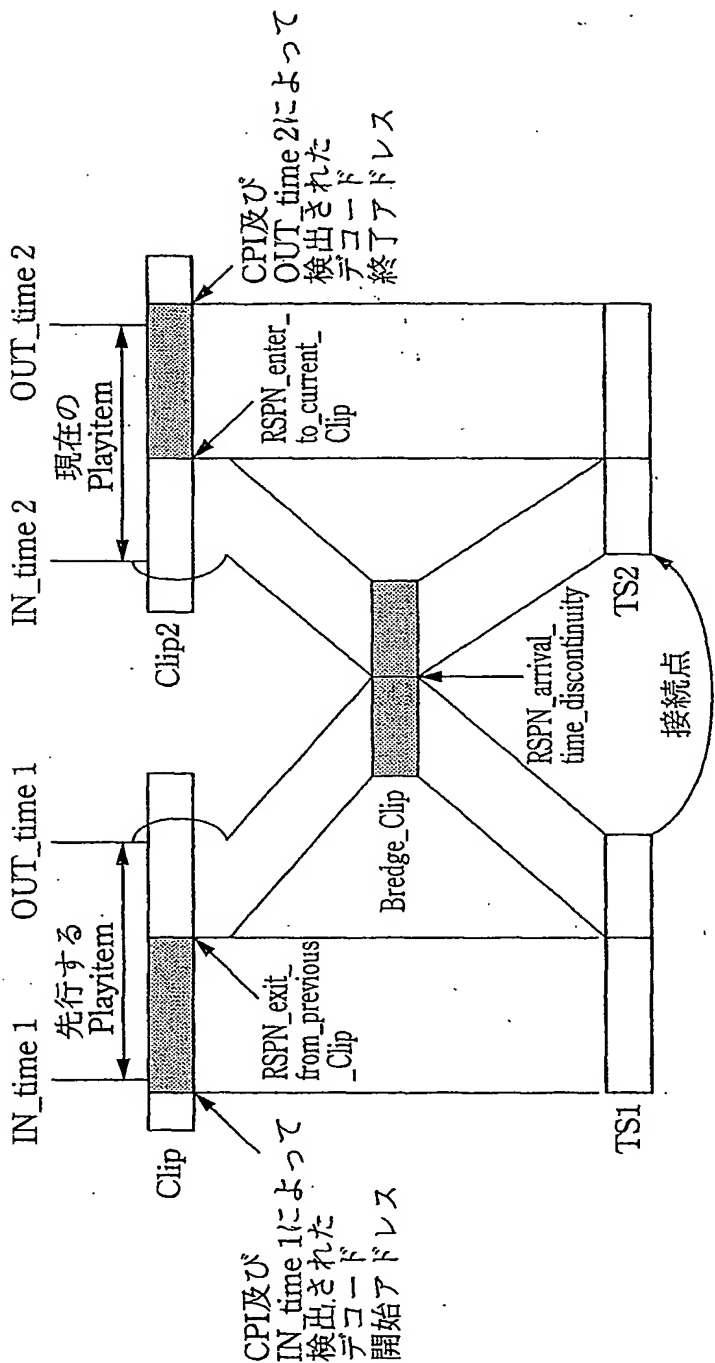


FIG.96

84/118

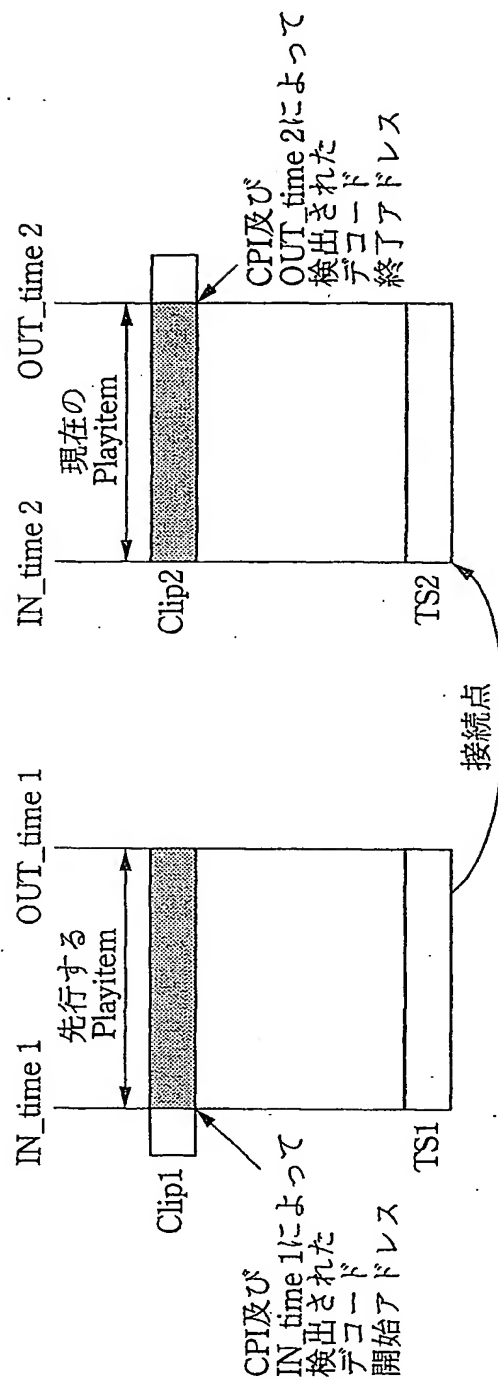


FIG.97

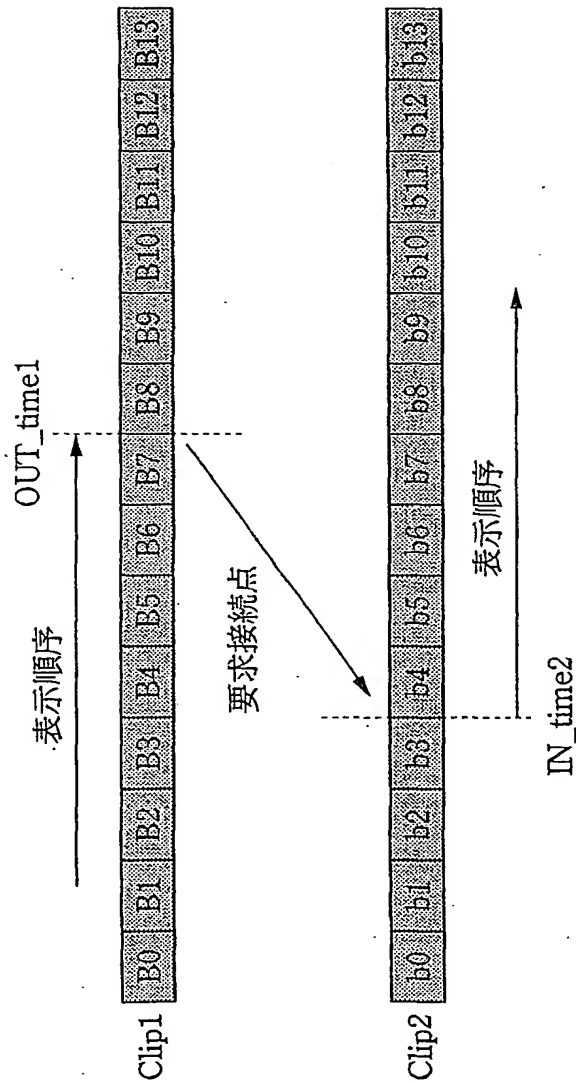


FIG.98

86/118

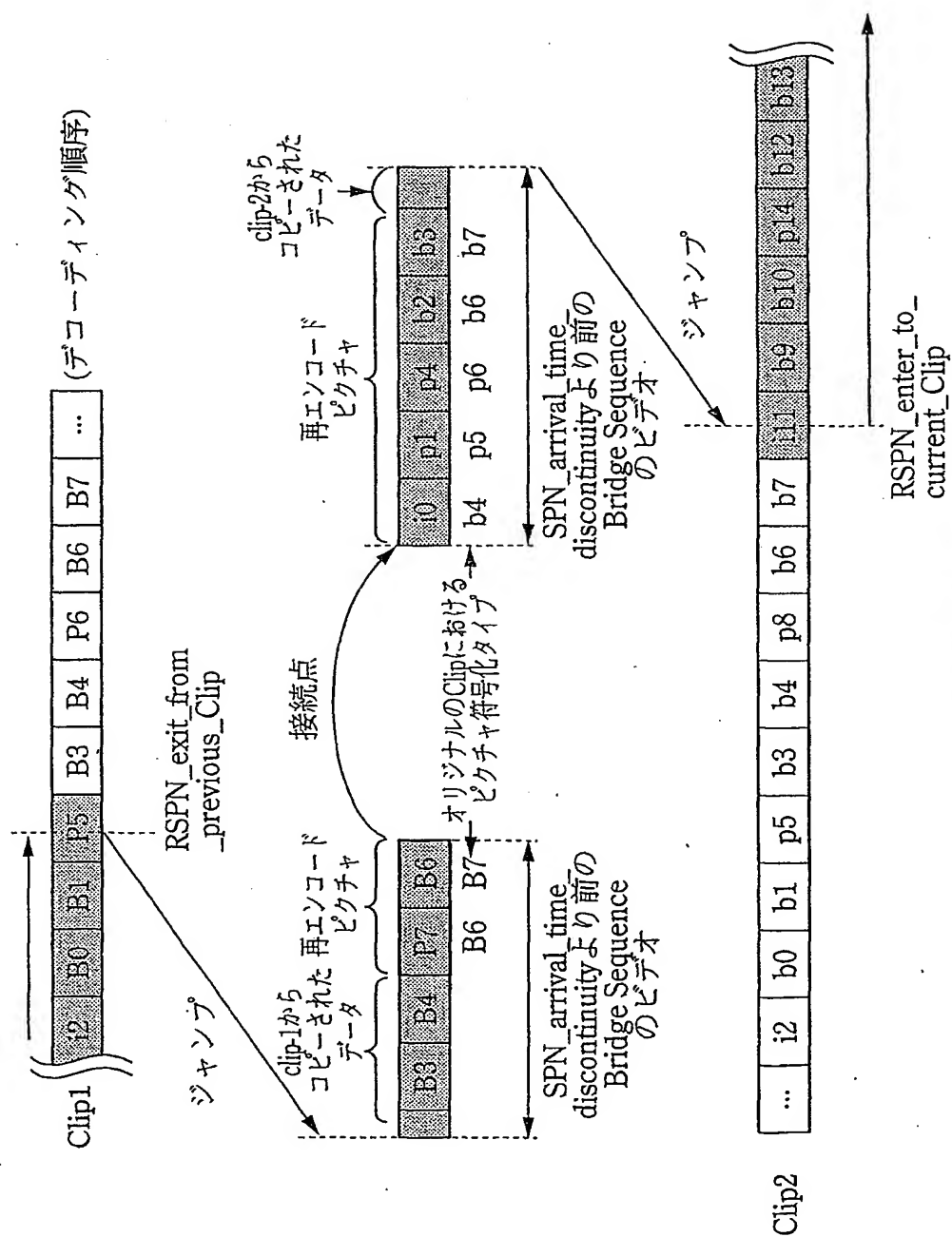


FIG.99

87/118

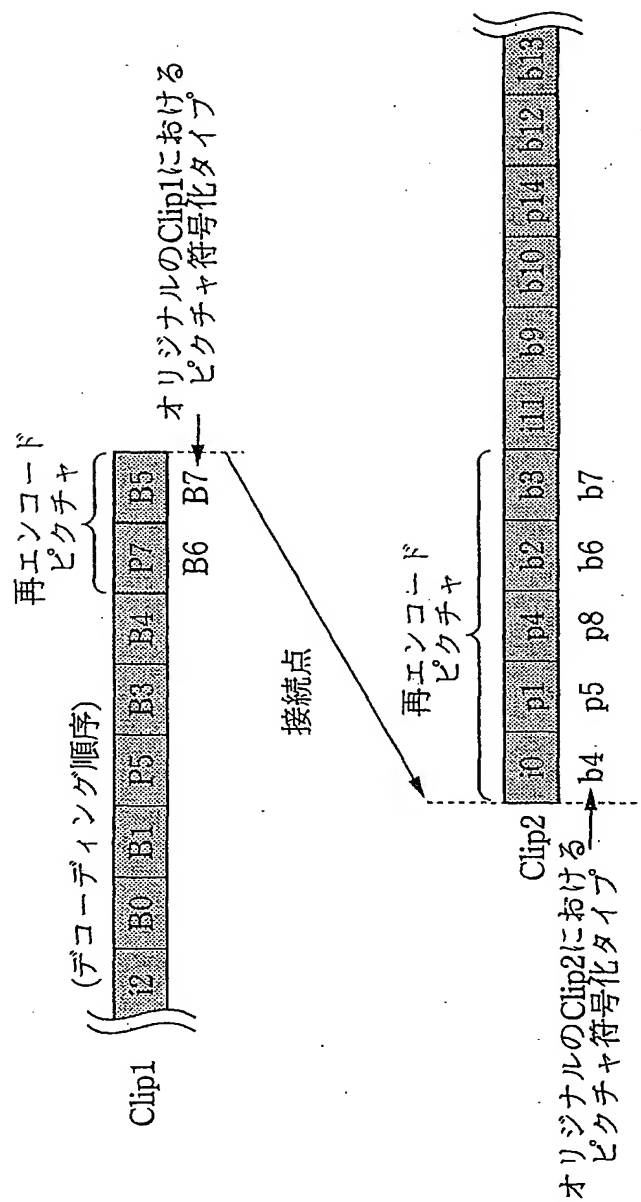


FIG.100

88/118

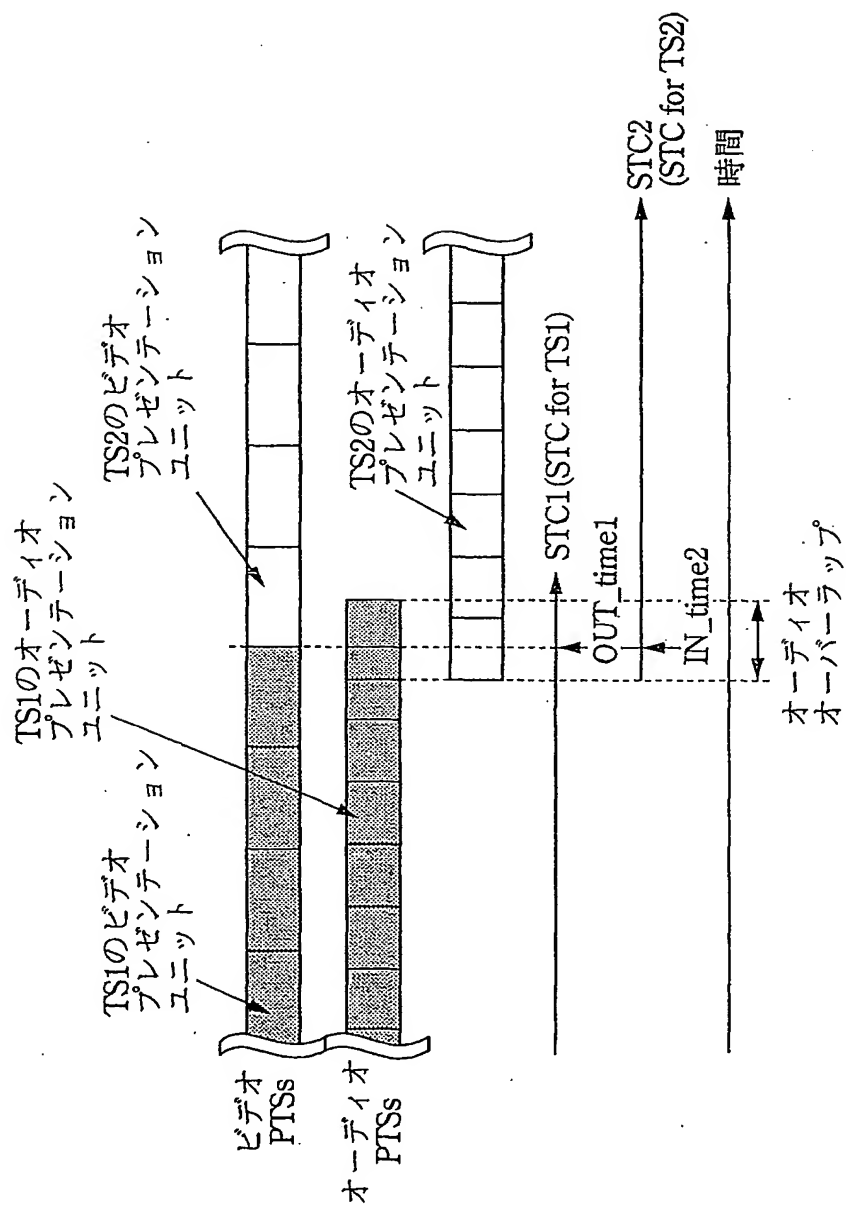


FIG.101

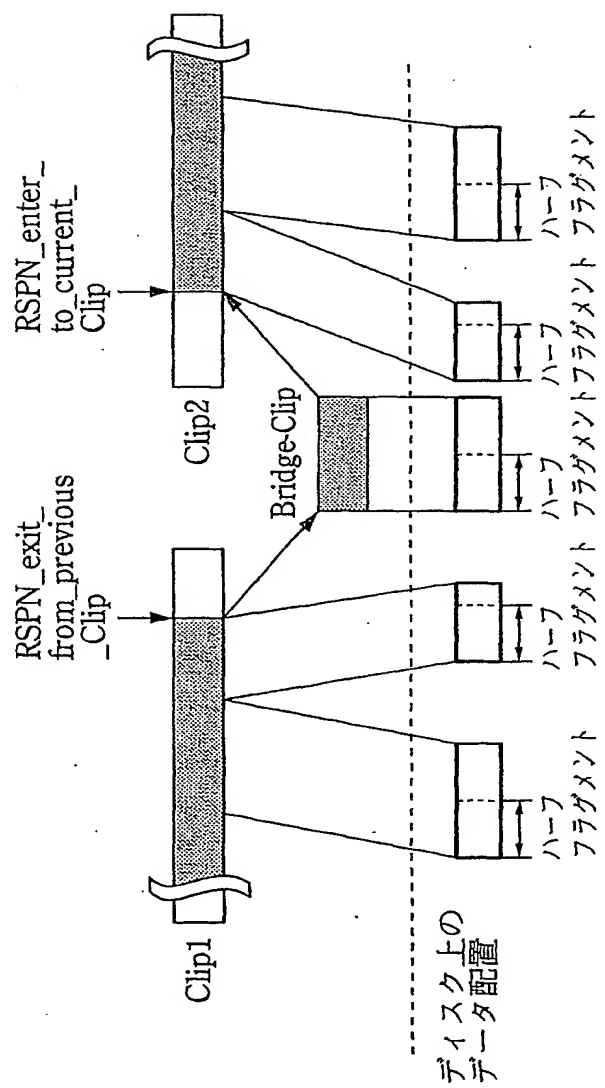


FIG.102

90/118

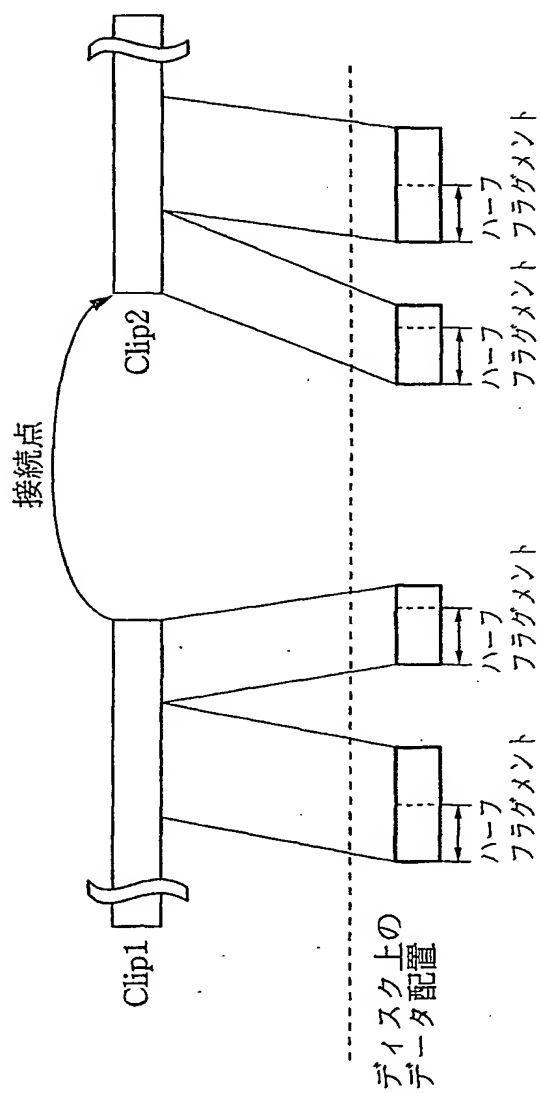


FIG.103

91/118

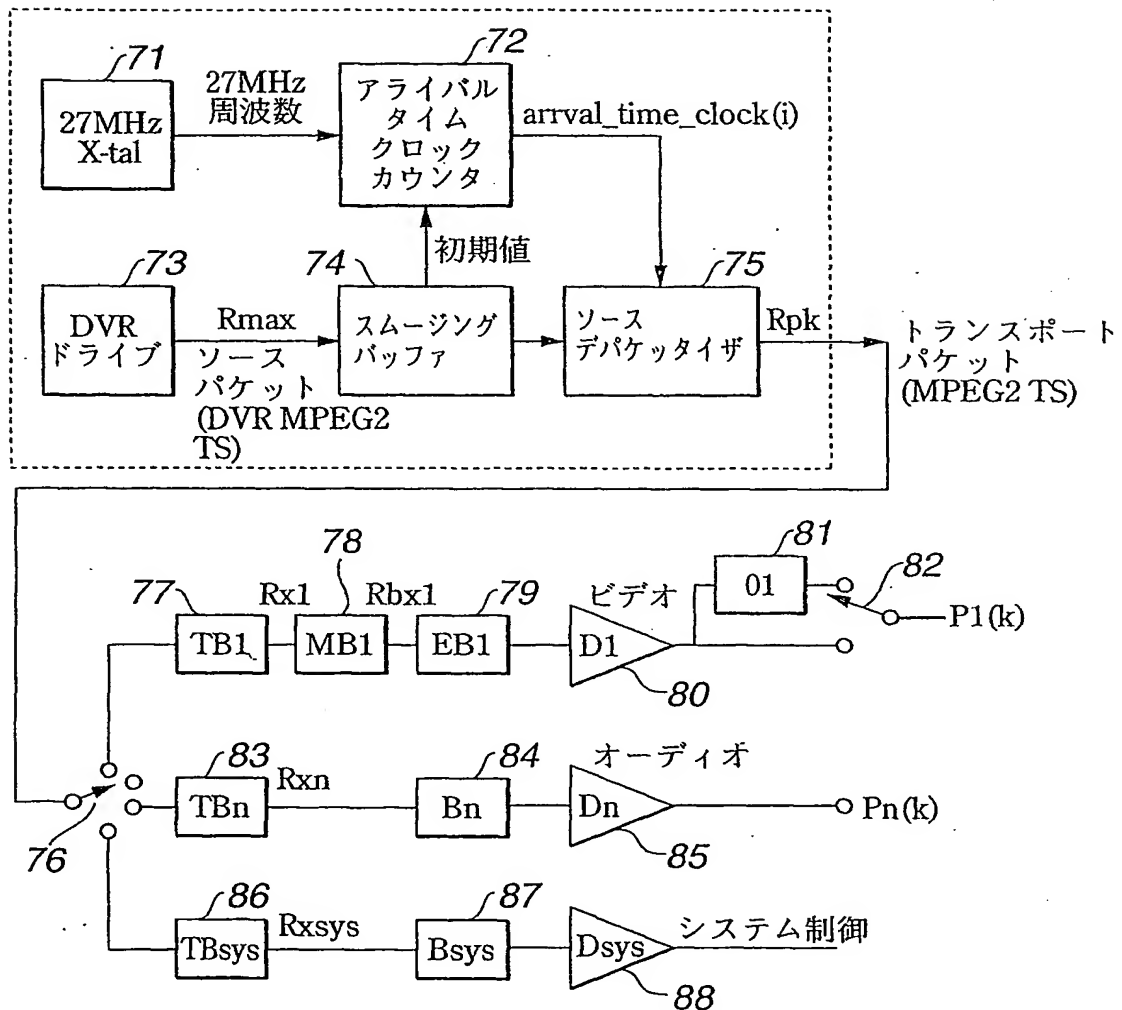


FIG.104

92/118

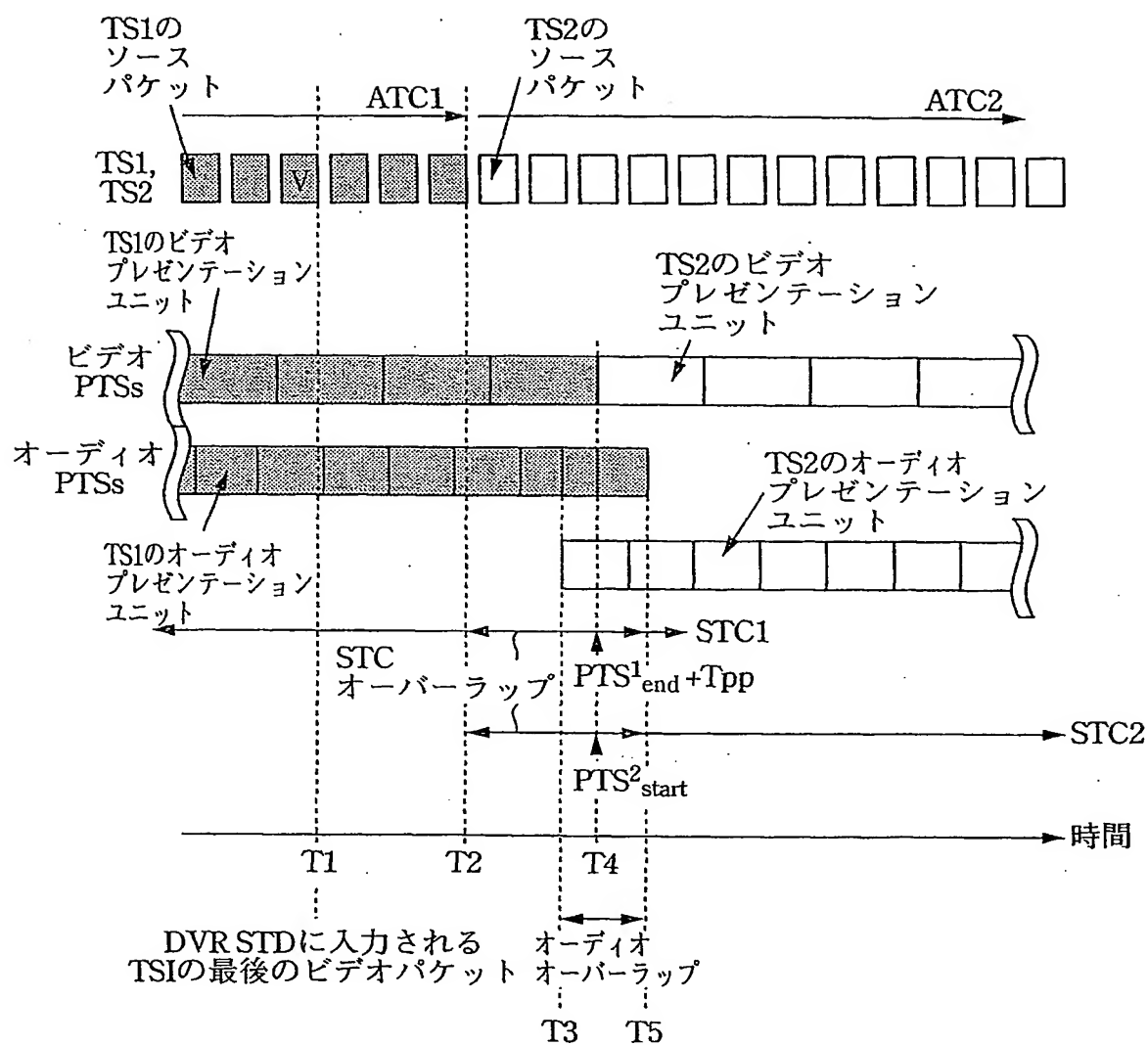


FIG.105

93/118

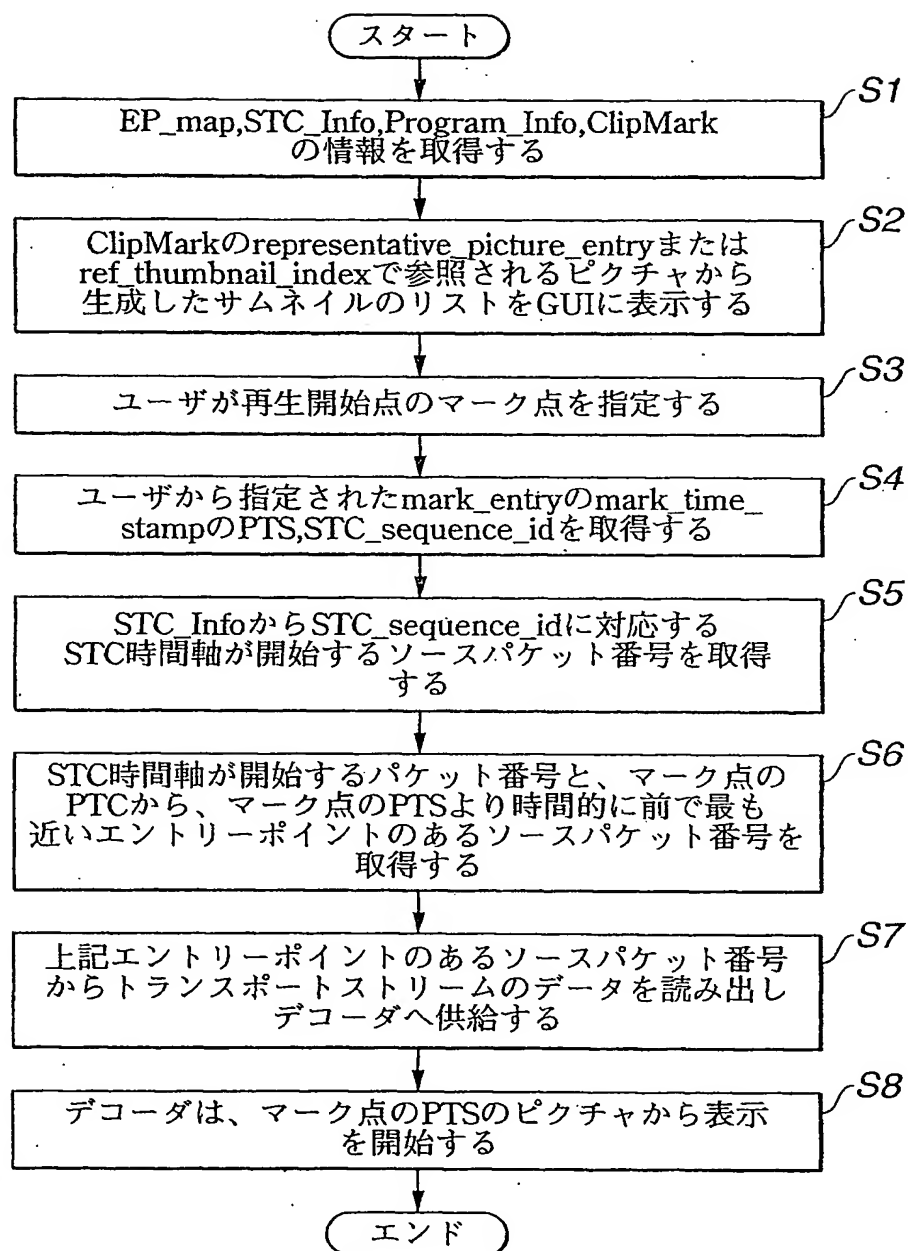


FIG.106

94/118

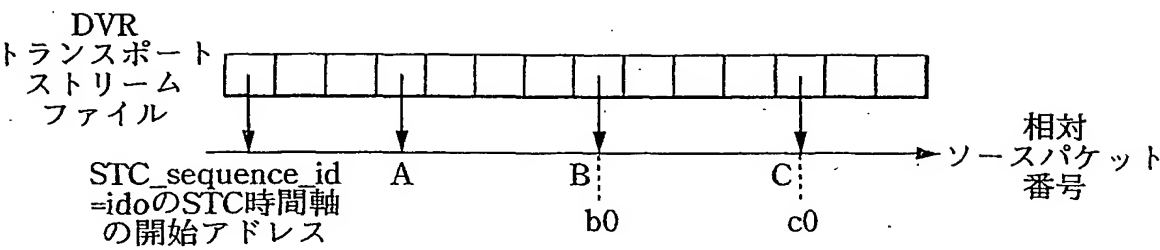


FIG.107

EP_map

RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

FIG.108

ClipMark

Mark_type	mark_entry		representative_picture_entry	
	Mark_Time_stamp	STC_sequence_id	Mark_Time_stamp	STC_sequence_id
...
0x92(scene start)	PTS(a1)	id0	PTS(a2)	id0
0x94(CM start)	PTS(b0)	id0	PTS(b0)	id0
0x95(CM end)	PTS(c0)	id0	PTS(c0)	id0
...

FIG.109

95/118

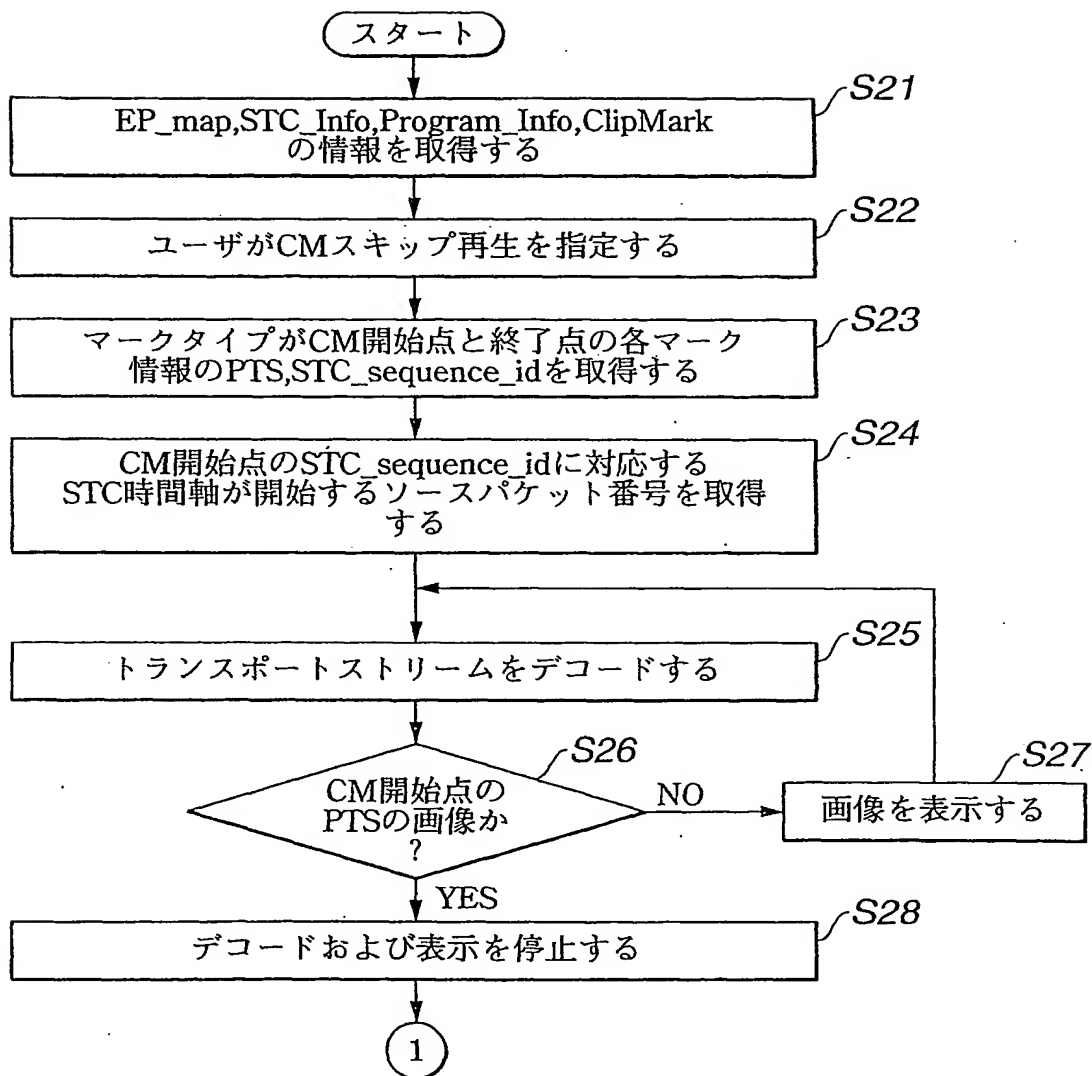


FIG.110

96/118

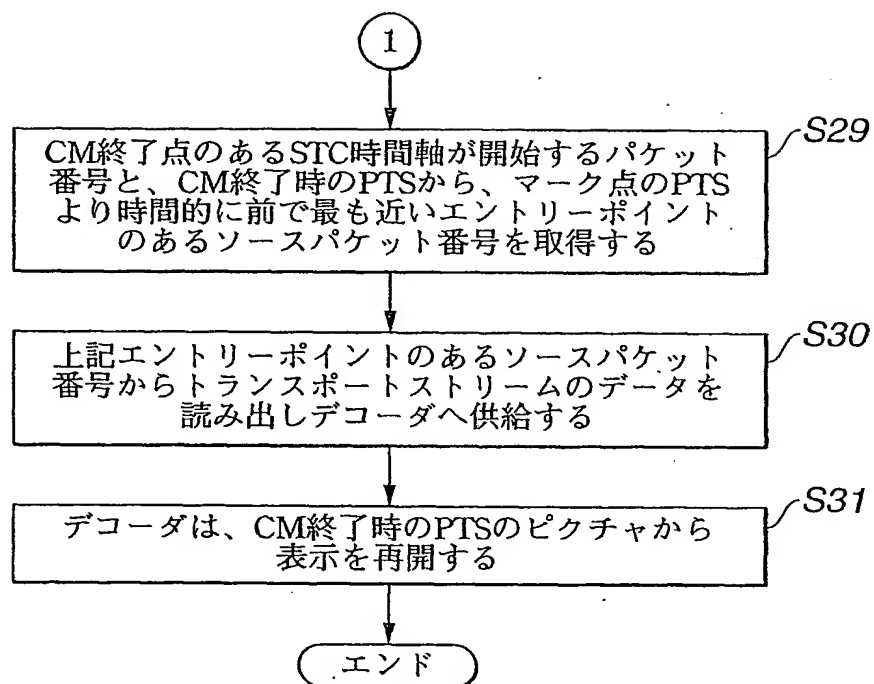


FIG.111

97/118

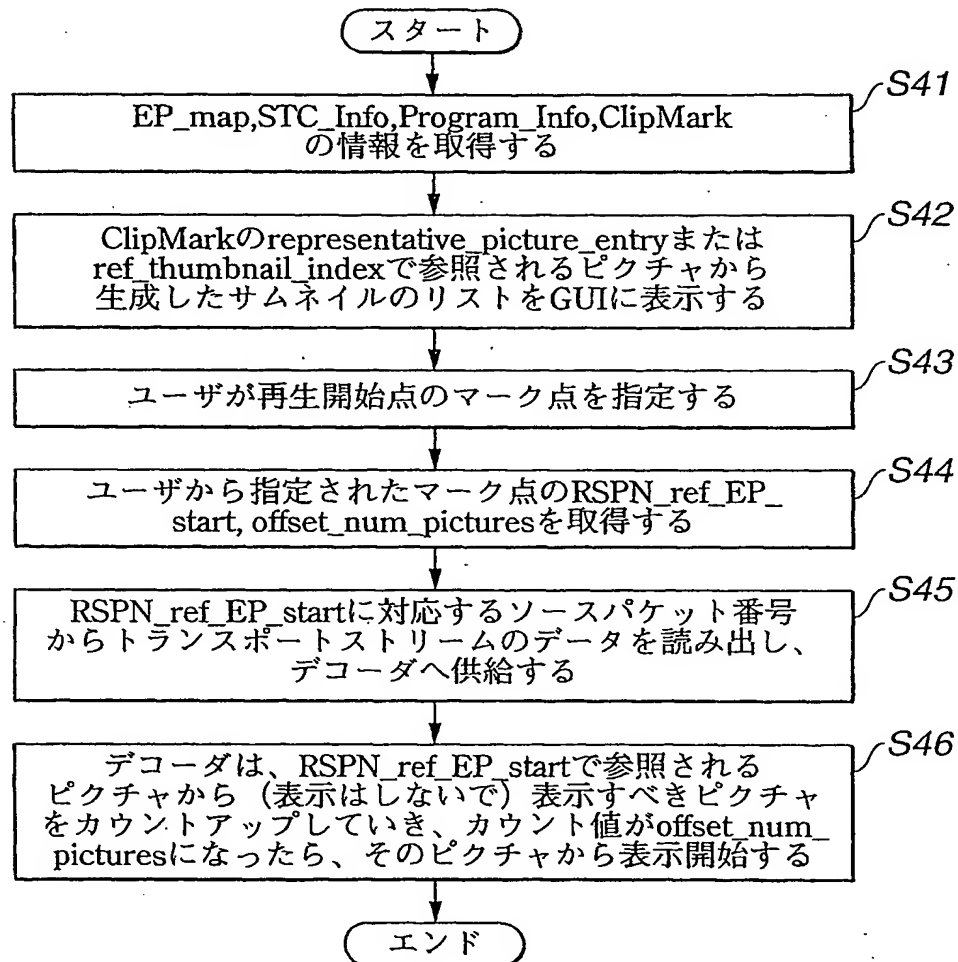


FIG.112

98/118

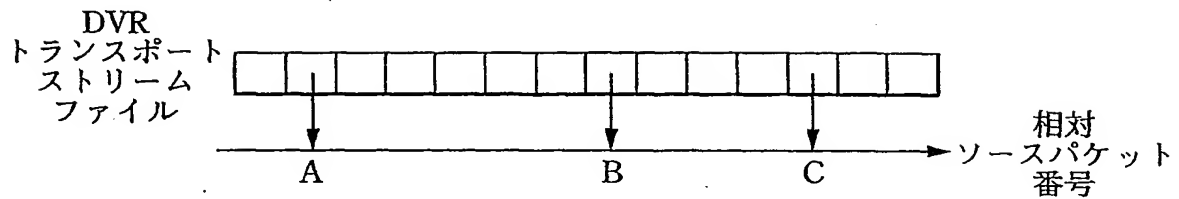


FIG.113

EP_map

RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

FIG.114

ClipMark

mark_type	mark_entry		representative_picture_entry	
	RSPN_ref_EP_start	offset_num_pictures	RSPN_ref_EP_start	offset_num_pictures
...
0x92(scene start)	A	M1	A	M2
0x94(CM start)	B	N1	B	N1
0x95(CM end)	C	N2	C	N2
...

FIG.115

99/118

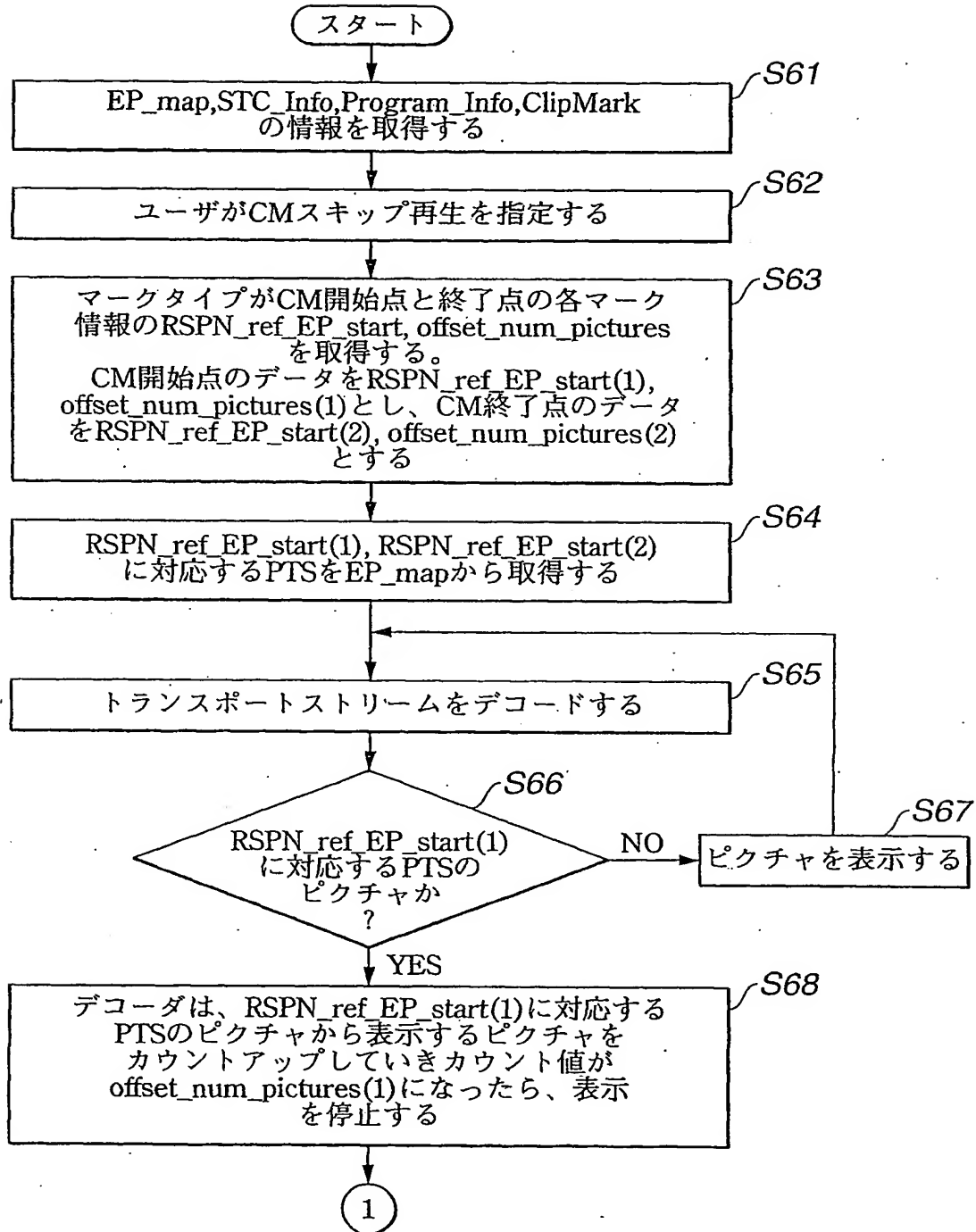


FIG.116

100/118

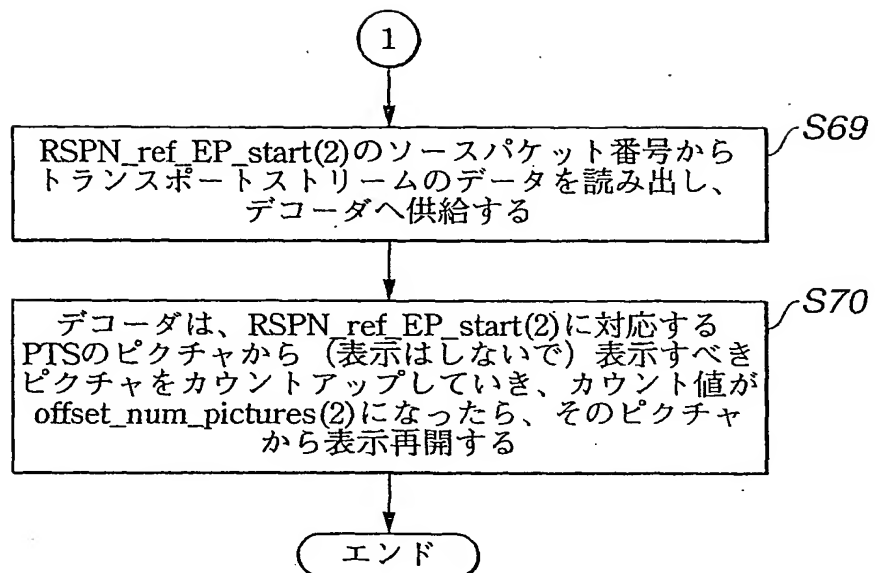


FIG.117

101/118

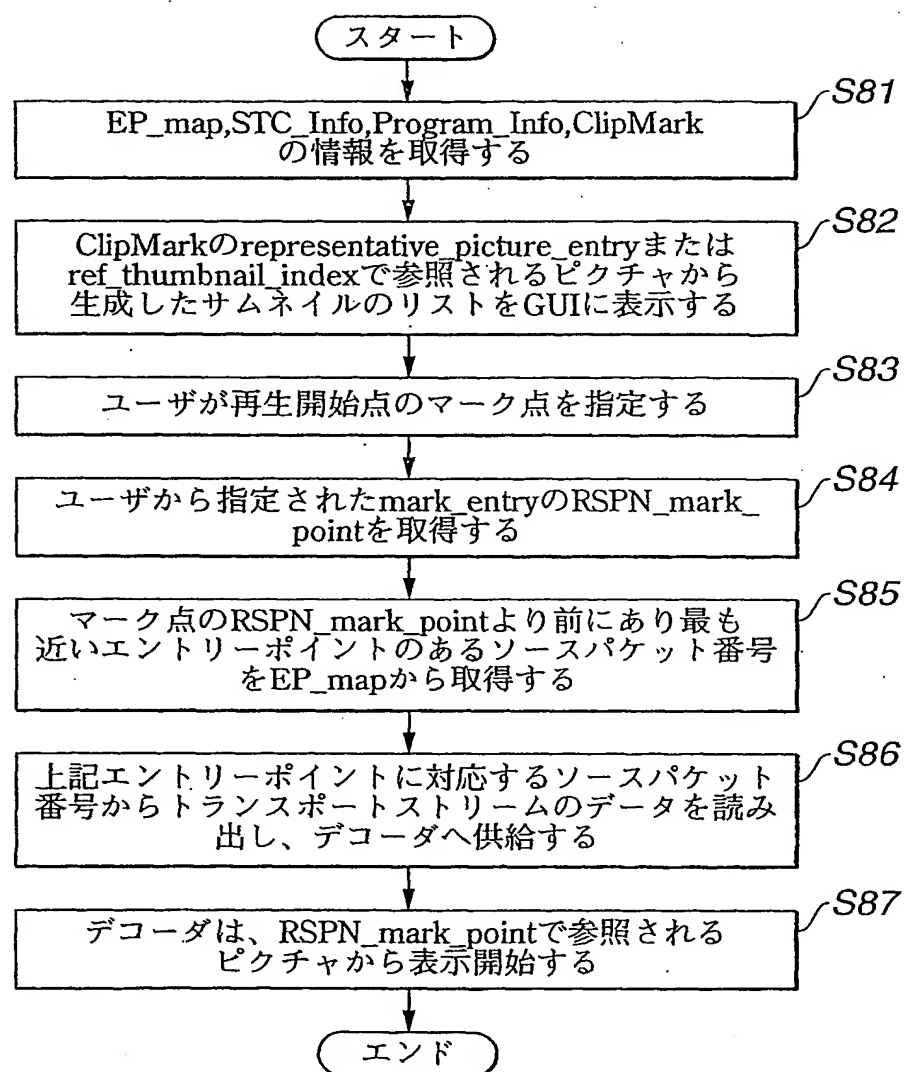


FIG.118

102/118

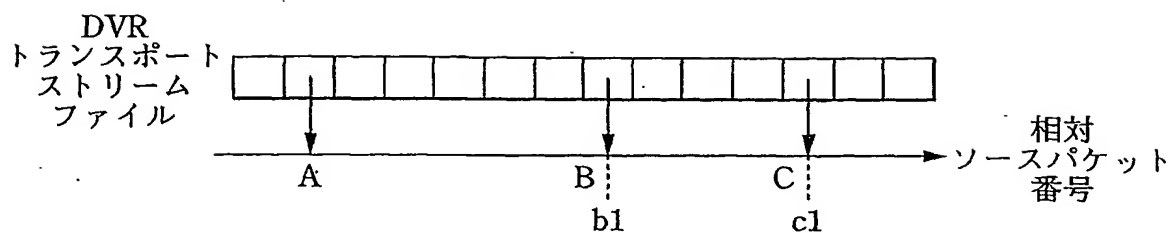


FIG.119

EP_map

RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

FIG.120

ClipMark

mark_type	mark_entry	representative_picture_entry
	RSPN_mark_point	RSPN_mark_point
...
0x92 (scene start)	a1	a2
0x94 (CM start)	b1	b1
0x95 (CM end)	c1	c1
...

FIG.121

103/118

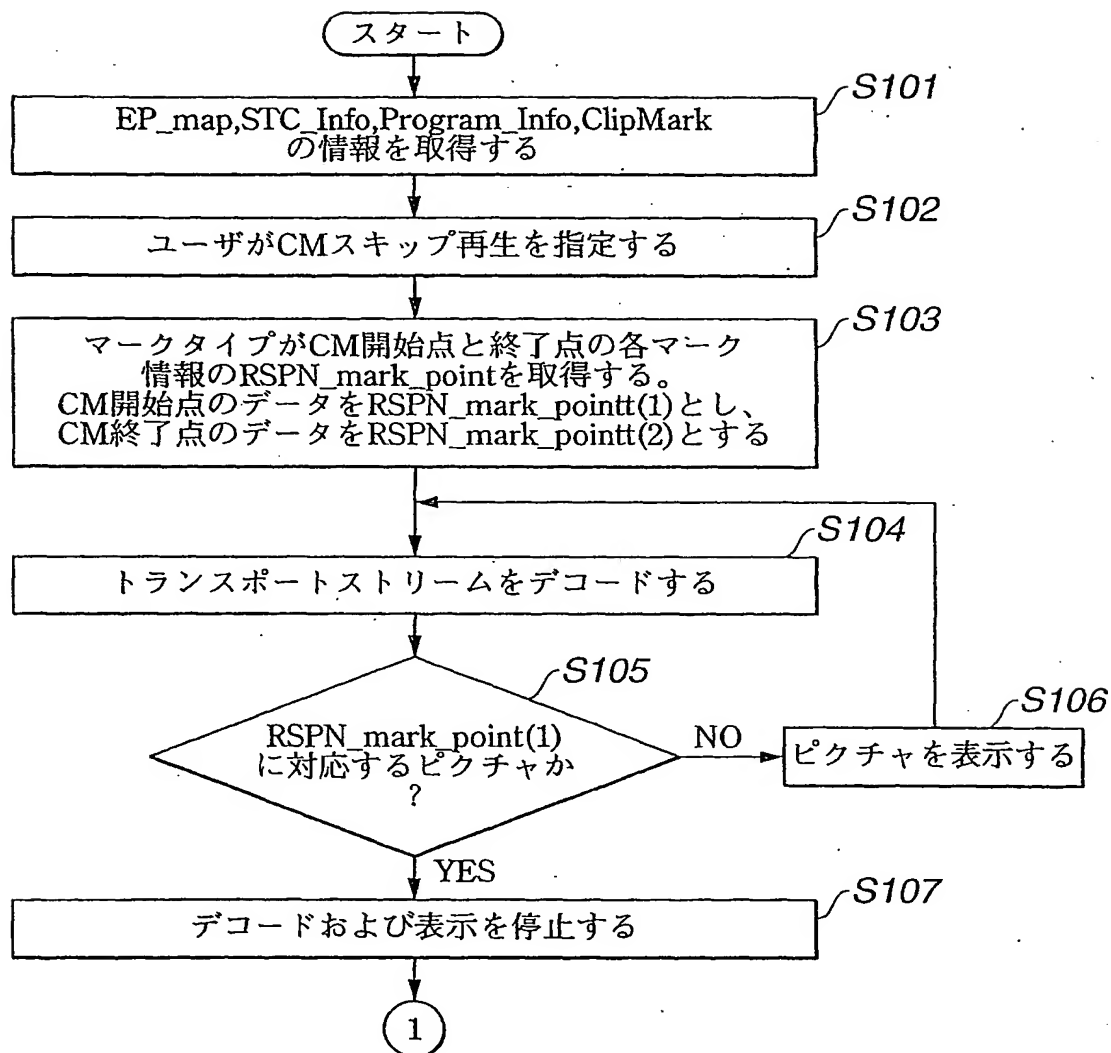


FIG.122

104/118

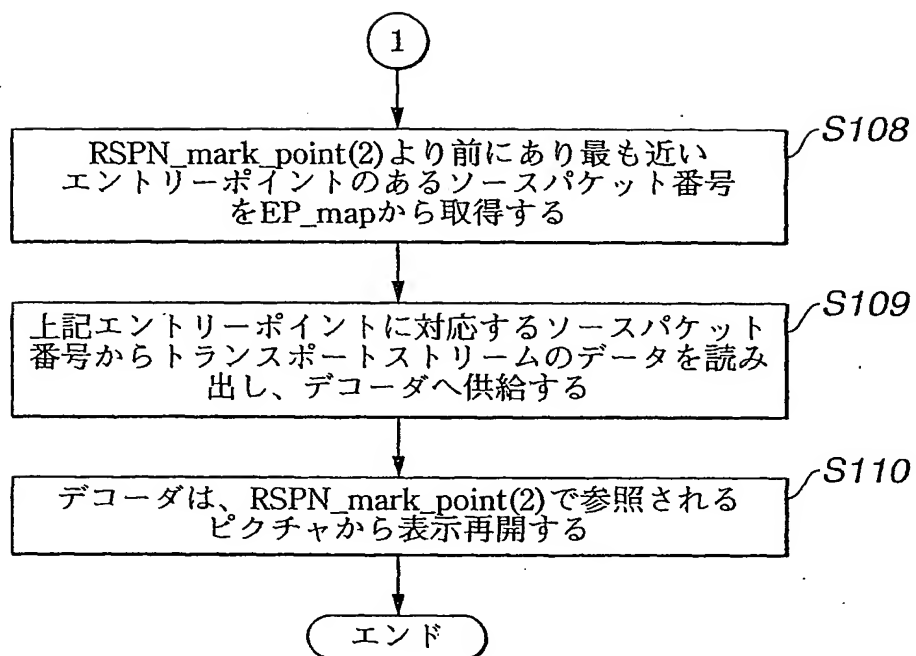


FIG.123

105/118

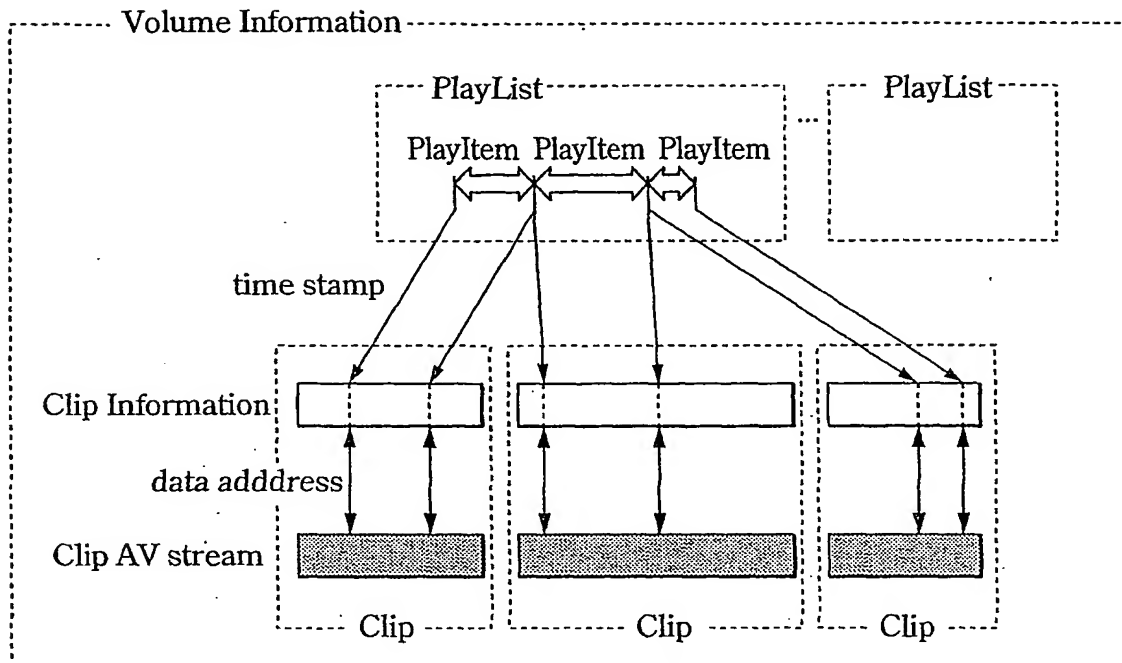


FIG.124

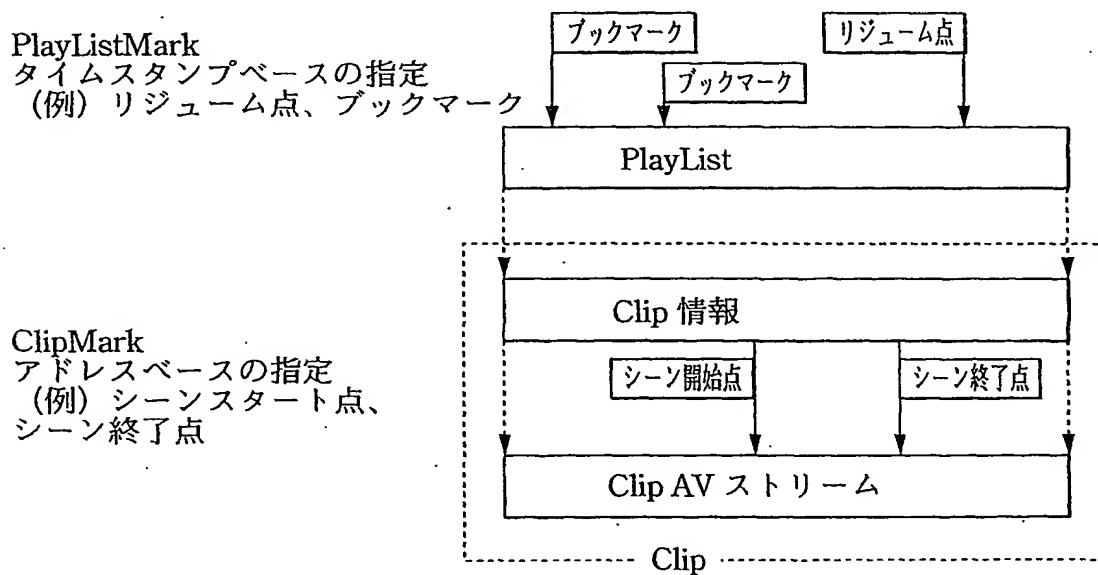


FIG.125

106/118

シンタクス	バイト数	略号
ClipMark0{		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<number_of_Clip_marks; i++){		
reserved	8	bslbf
mark_type	8	bslbf
RSPN_mark	32	uimsbf
reserved	32	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

FIG.126

シンタクス	バイト数	略号
ClipMark0{		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<number_of_Clip_marks; i++){		
reserved	8	bslbf
mark_type	8	bslbf
RSPN_ref_EP_start	32	uimsbf
offset_num_pictures	32	uimsbf
ref_thumbnail_index	16	uimsbf
}		
}		

FIG.127

107/118

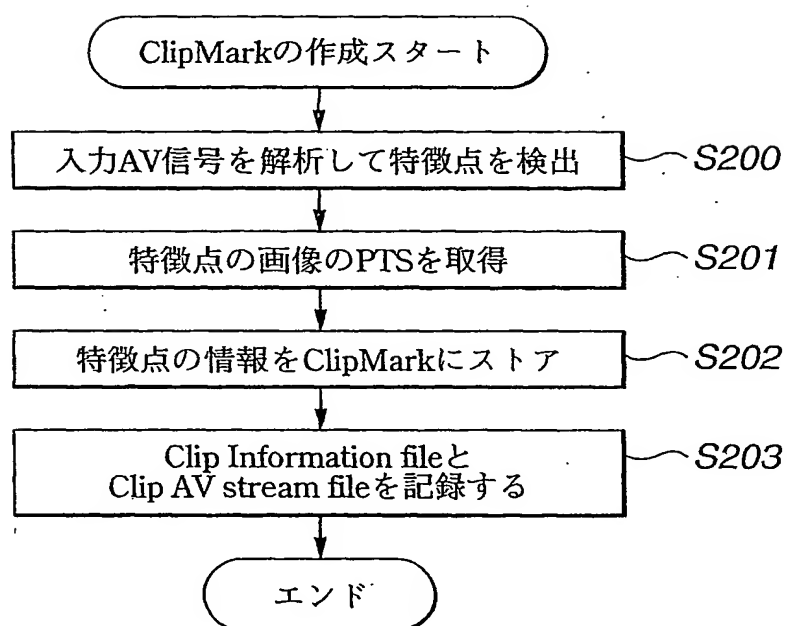


FIG.128

108/118

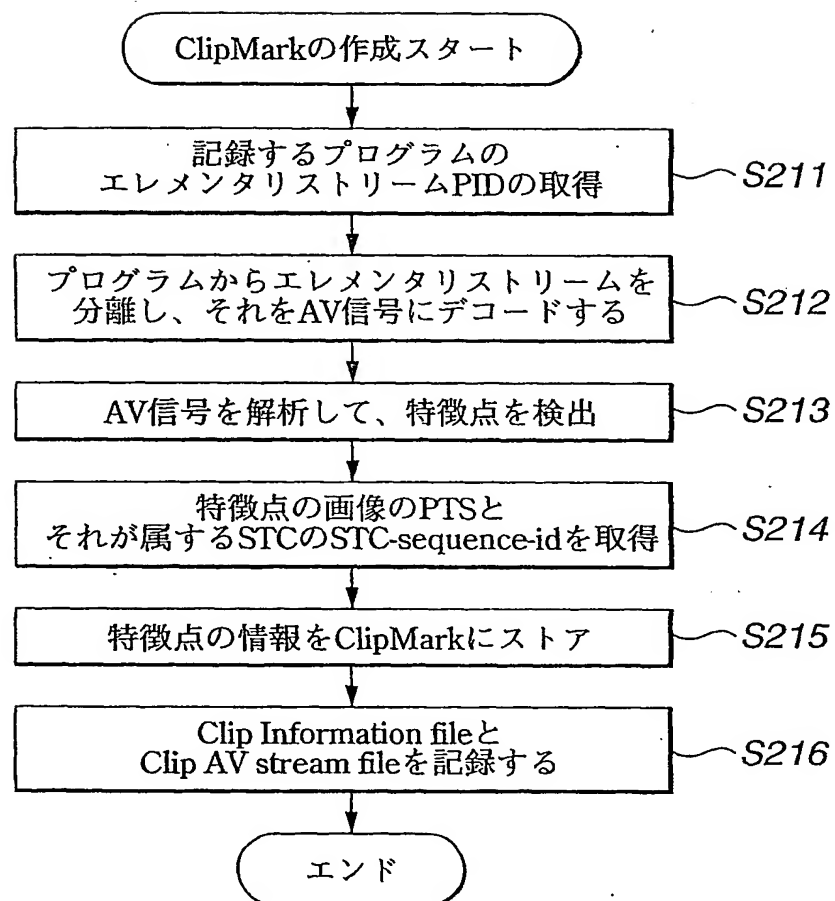


FIG.129

109/118

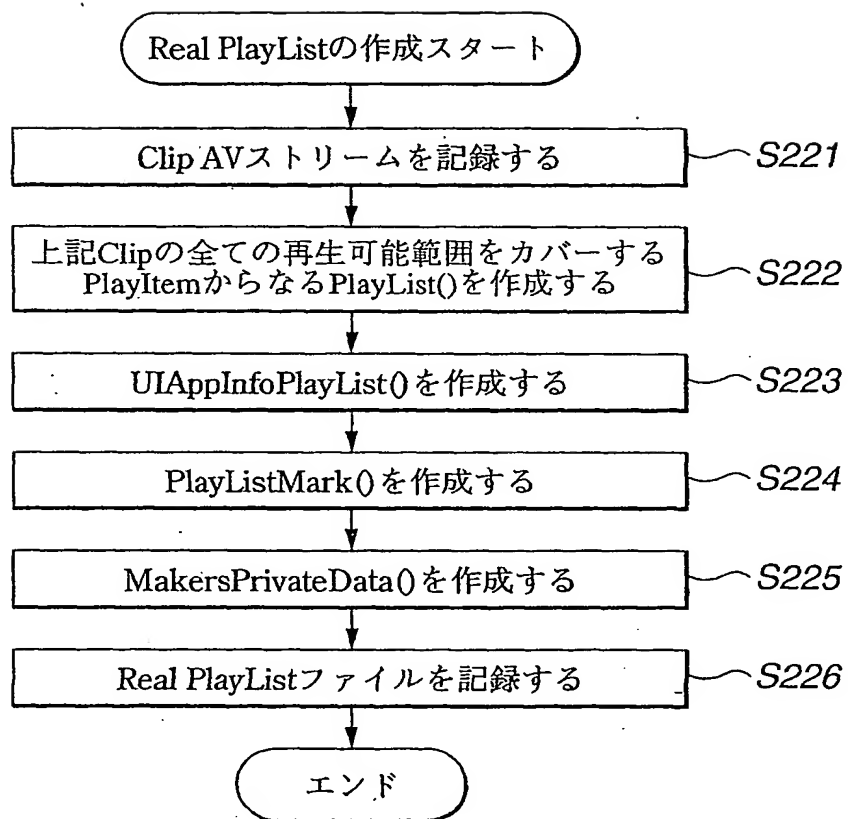


FIG.130

110/118

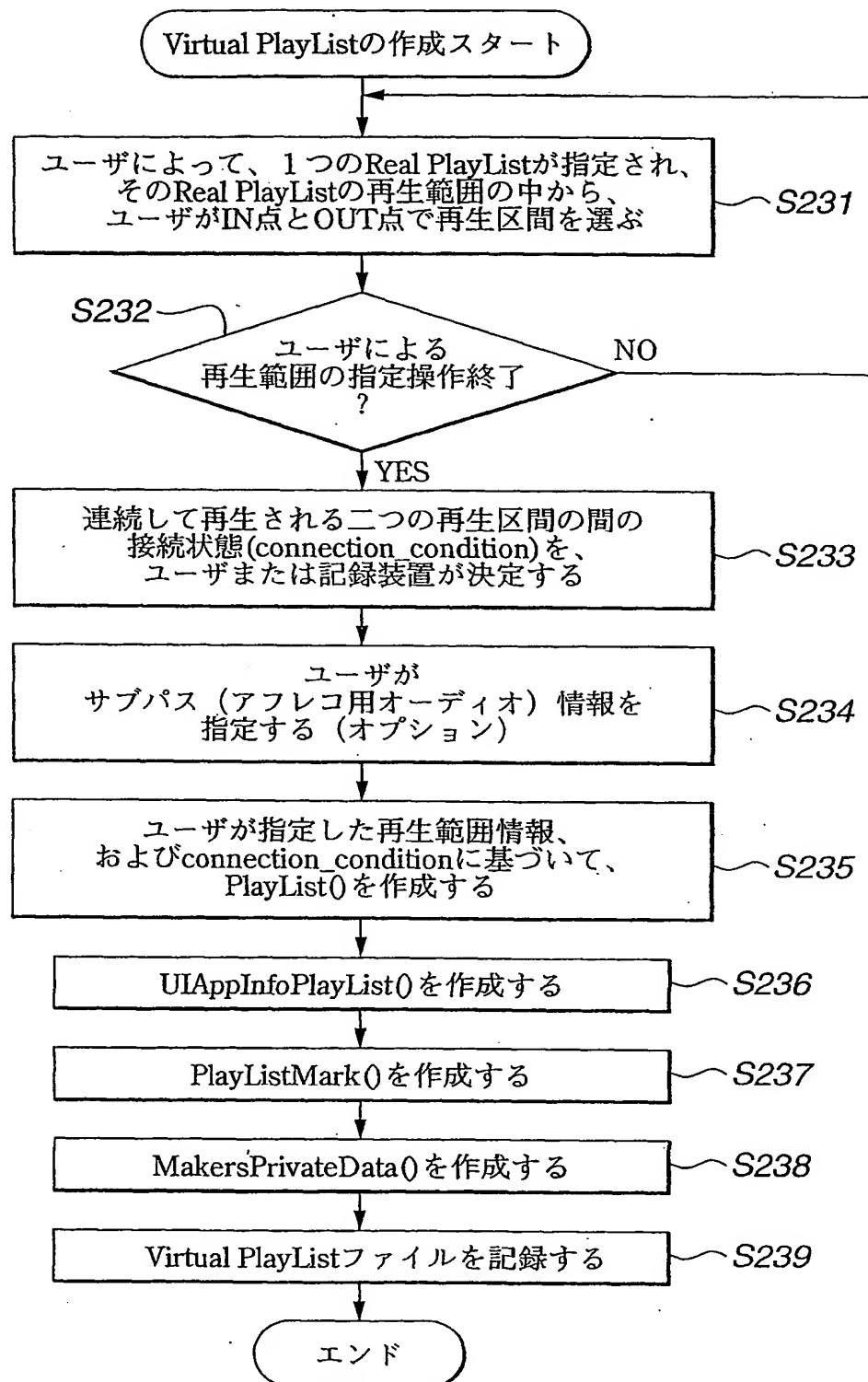


FIG.131

111/118

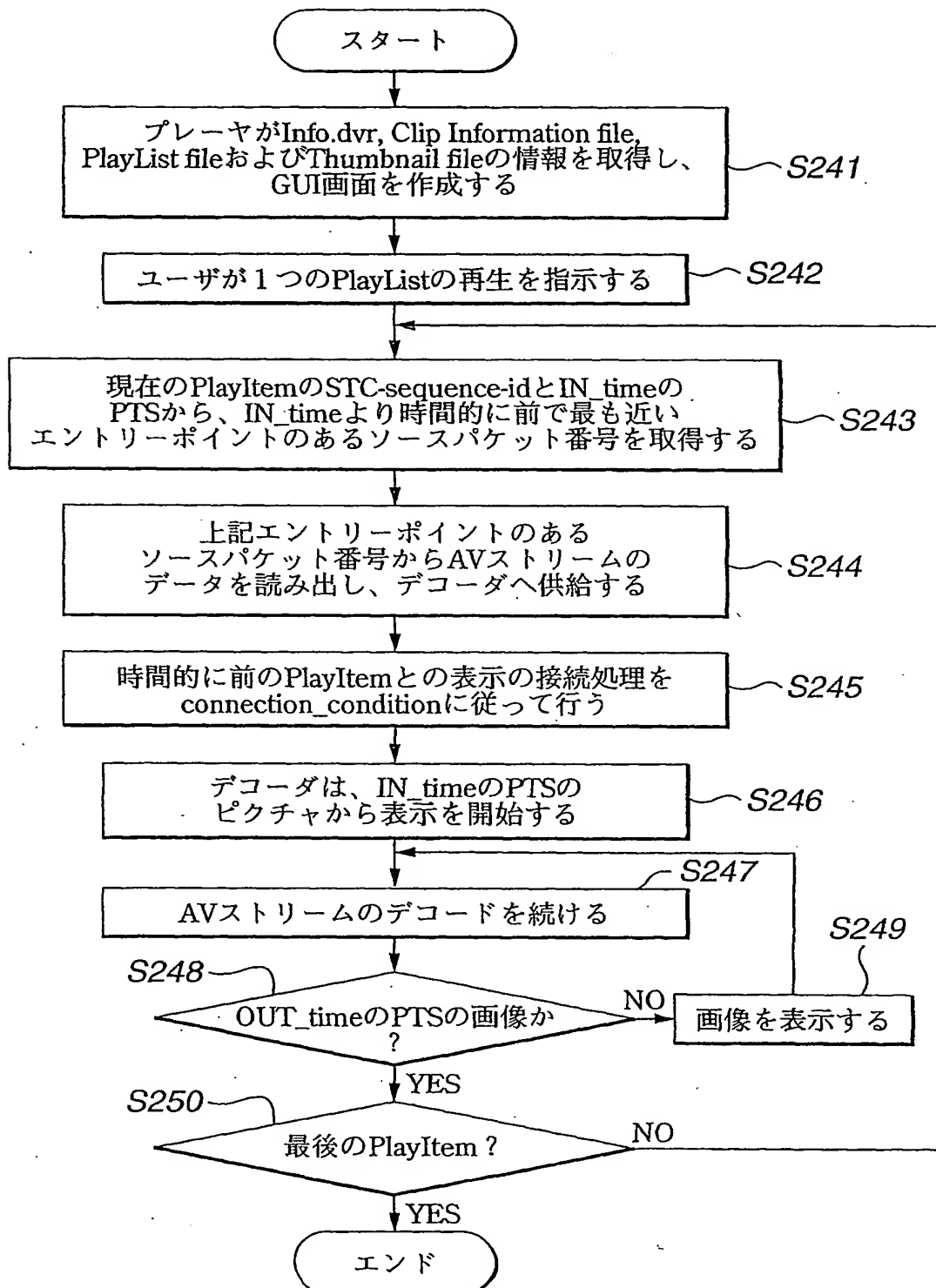


FIG.132

112/118

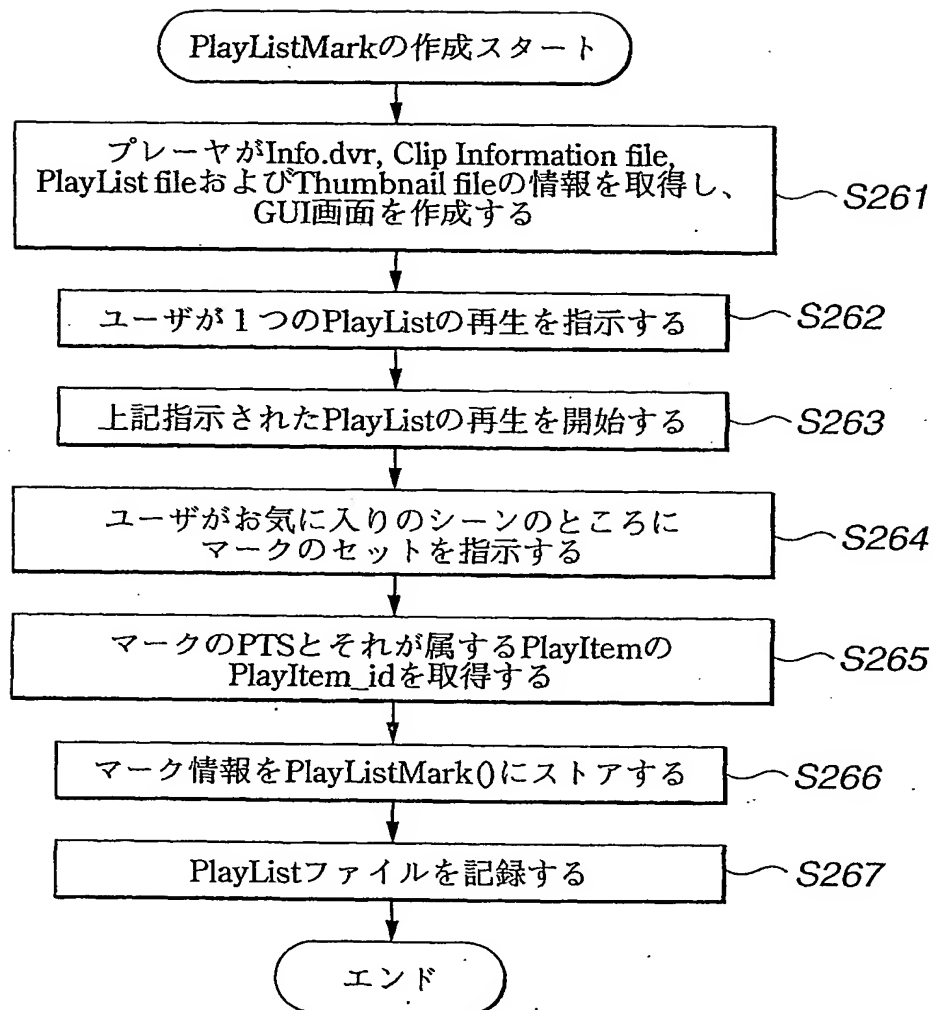


FIG.133

113/118

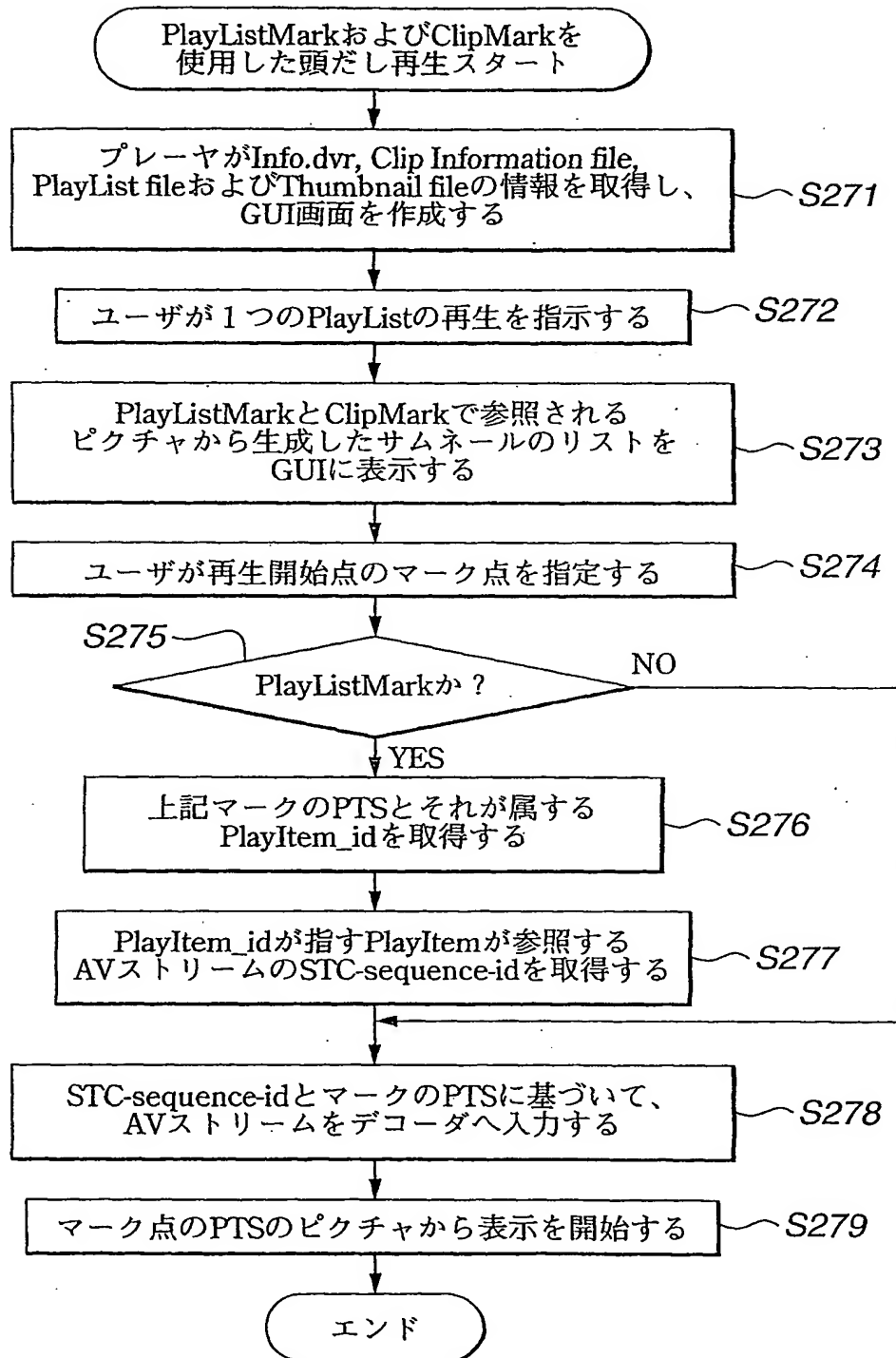


FIG.134

114/118

シンタクス	バイト数	略号
PlayListMark0{		
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for (i=0;i<number_of_PlayList_marks;i++){		
mark_invalid_flag	1	uimsbf
mark_type	7	uimsbf
mark_name_length	8	uimsbf
ref_to_PlayItem_id	16	uimsbf
mark_time_stamp	32	uimsbf
entry_ES_PID	16	uimsbf
ref_to_thumbnail_index	16	uimsbf
mark_name	8*32	bslbf
}		
}		

FIG.135

115/118

値	意味	ノート
0x00	Resume-mark	再生リジュームポイント。PlayListMark()において定義される再生リジュームポイントの数は、0または1でなければならない。
0x01	Book-mark	PlayListの再生エントリーポイント。このマークは、ユーザがセットすることができ、例えば、お気に入りのシーンの開始点を指定するマークに使う。このマークは、PlayListMark()に複数あっても良い。
0x02	Chapter-mark	ユーザは、PlayListの中で1つのチャプターがこのマークから開始することを意図している。ユーザがセットすることができる。このマークは、PlayListMark()に複数あっても良い。
0x03	Skip-start-mark	PlayListMarkの中に1つのSkip-start-markがセットされる場合、そのSkip-start-markのエントリーの直後に1つのSkip-end-markがセットされなければならない。Skip-start-markのタイムスタンプからSkip-end-markのタイムスタンプまで、ユーザは、PlayListの再生をスキップすることを意図している。 Skip-start-markとSkip-end-markは、同じref_to_PlayItem_idを持つ。また、Skip-start-markとSkip-end-markは、もしentry_ES_PIDが0xFFFFでないならば、同じentry_ES_PIDの値を持つ。 ユーザがセットすることができるマークであり、このマークは、PlayListMark()に複数あっても良い。
0x04	Skip-end-mark	
0x05-0x3F	Reserved for future use	Reserved for PlayListMark
0x40-0x7F	Reserved for ClipMark	

FIG.136

116/118

シンタクス	バイト数	略号
ClipMark(){		
length	32	uimsbf
maker_ID	16	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<number_of_Clip_marks; i++){		
mark_invalid_flag	1	uimsbf
mark_type	7	uimsbf
ref_to_STC_id	8	uimsbf
mark_time_stamp	32	uimsbf
entry_ES_PID	16	uimsbf
ref_to_thumbnail_index	16	uimsbf
representative_picture_time_stamp	32	uimsbf
}		
}		

FIG.137

117/118

Mark_type	意味	ノート
0x00- 0x3F	reserved for future use	Reserved for PlayListMark
0x40	Scene-start-mark	シーンの開始ポイントを示すマーク点
0x41- 0x5F	Reserved for common ClipMark	
0x60- 0x7F	Maker defined ClipMark	maker_IDによって示されるメーカーが 自由に意味を定義できる

FIG.138

118/118

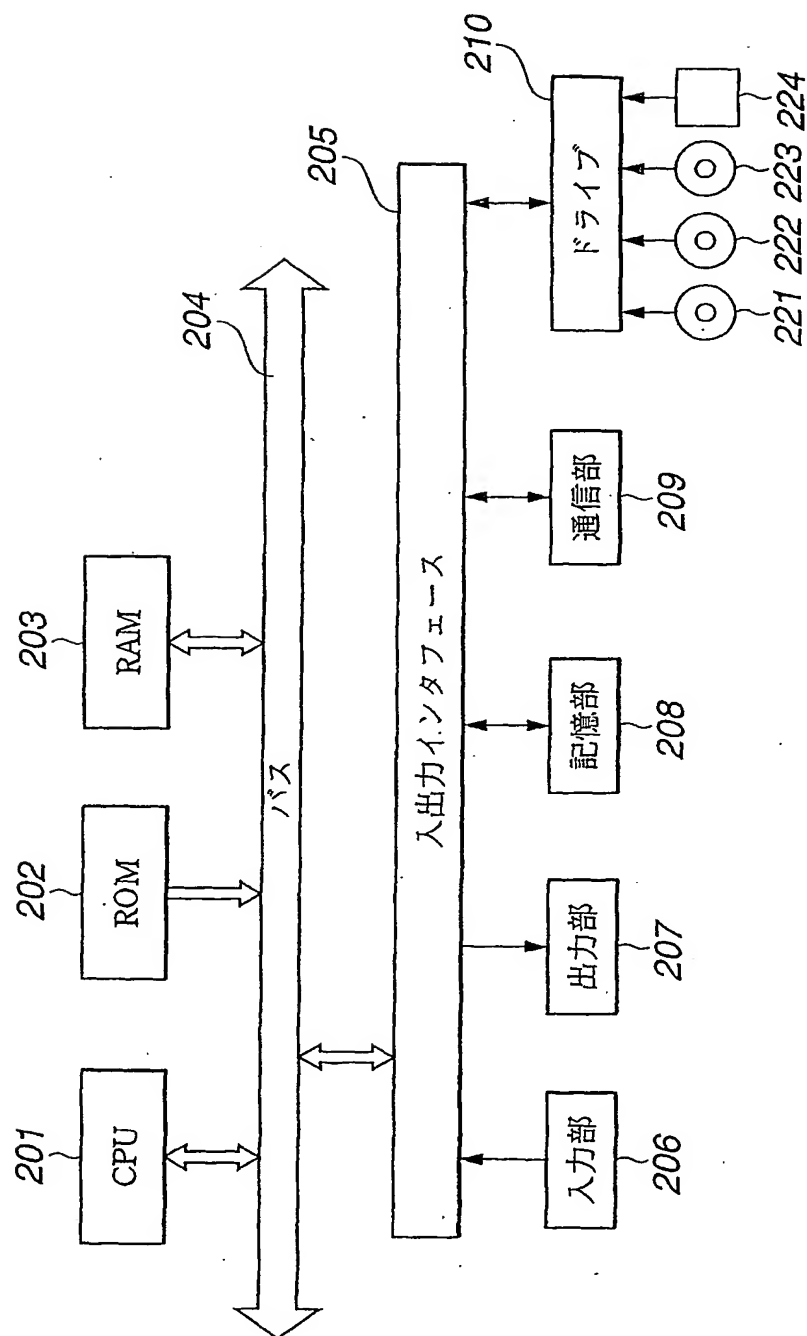


FIG.139

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP01/03414

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl⁷ H04N 5/93, G11B 20/10

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl⁷ H04N 5/76-5/956, G11B 20/10-20/12

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho 1922-1996 Toroku Jitsuyo Shinan Koho 1994-2001
 Kokai Jitsuyo Shinan Koho 1971-2001 Jitsuyo Shinan Toroku Koho 1996-2001

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P, A	JP, 2000-341646, A (Sony Corporation), 08 December, 2000 (08.12.00), Full text; Figs. 1 to 10 (Family: none)	1-23
X	JP, 11-273227, A (NEC Software Kobe Ltd.), 08 October, 1999 (08.10.99), Full text; Figs. 1 to 3 (Family: none)	1-23
A	JP, 10-290432, A (Matsushita Electric Ind. Co., Ltd.), 27 October, 1998 (27.10.98), Full text; Figs. 1 to 10 (Family: none)	1-23

☐ Further documents are listed in the continuation of Box C.☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance
 "E" earlier document but published on or after the international filing date
 "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
 "O" document referring to an oral disclosure, use, exhibition or other means
 "P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
 "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
 "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
 "&" document member of the same patent family

Date of the actual completion of the international search
 12 June, 2001 (12.06.01)

Date of mailing of the international search report
 19 June, 2001 (19.06.01)

Name and mailing address of the ISA/
 Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類 (国際特許分類 (IPC))		
Int Cl ⁷ H04N 5/93, G11B 20/10		
B. 調査を行った分野		
調査を行った最小限資料 (国際特許分類 (IPC))		
Int Cl ⁷ H04N 5/76-5/956, G11B 20/10-20/12		
最小限資料以外の資料で調査を行った分野に含まれるもの		
日本国実用新案公報 1922-1996年 日本国公開実用新案公報 1971-2001年 日本国登録実用新案公報 1994-2001年 日本国実用新案登録公報 1996-2001年		
国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)		
C. 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
P, A	JP, 2000-341646, A (ソニー株式会社) 08.12月.2000 (08.12.00) 全文、第1-10図 (ファミリーなし)	1-23
X	JP, 11-273227, A (神戸日本電気ソフトウェア株式会社) 08.10月.1999 (08.10.99) 全文、第1-3図 (ファミリーなし)	1-23
A	JP, 10-290432, A (松下電器産業株式会社) 27.10月.1998 (27.10.98) 全文、第1-10図 (ファミリーなし)	1-23
<input type="checkbox"/> C欄の続きにも文献が列挙されている。 <input type="checkbox"/> パテントファミリーに関する別紙を参照。		
* 引用文献のカテゴリー 「A」 特に関連のある文献ではなく、一般的技術水準を示すもの 「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの 「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す) 「O」 口頭による開示、使用、展示等に言及する文献 「P」 国際出願日前で、かつ優先権の主張の基礎となる出願日の後に公表された文献 「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの 「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの 「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの 「&」 同一パテントファミリー文献		
国際調査を完了した日	12.06.01	国際調査報告の発送日
国際調査機関の名称及びあて先 日本国特許庁 (ISA/J P) 郵便番号100-8915 東京都千代田区霞が関三丁目4番3号		特許庁審査官 (権限のある職員) 鈴木 明 電話番号 03-3581-1101 内線 3541